The Dissertation Committee for Dustin Ray Reinert Certifies that this is the approved version of the following dissertation:

Investigation of Stochastic Radiation Transport Methods in Random Heterogeneous Mixtures

Committee:

Steven Biegalski, Co-Supervisor

Erich Schneider, Co-Supervisor

Sheldon Landsberger

John Howell

Timothy Donovan

Investigation of Stochastic Radiation Transport Methods in Random Heterogeneous Mixtures

by

Dustin Ray Reinert, B.S.; M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2008

Dedication

To my parents

Acknowledgements

I would like to thank my advisors, Dr. Steven Biegalski and Dr. Erich Schneider, for their assistance and support throughout the duration of this project. Their patience and encouragement has helped me to overcome many obstacles and I have learned a great deal from them. Dr. Biegalski has served as a mentor to me for several years and it has been a privilege to work with him during the course of my undergraduate and graduate education. His door is always open, and I am continually impressed with his commitment to his students' success. Since coming on board with the nuclear program at UT two years ago, Dr. Schneider has made a big mark on all of his students, and I am very grateful that he has been supportive of my work.

Funding for this project was provided by the Naval Nuclear Propulsion Fellowship program sponsored by the Naval Reactors Division of the US Department of Energy. Special thanks is due to Dr. Tim Donovan of Knolls Atomic Power Laboratory for serving on my committee. His encouragement and insight into Monte Carlo method development was a valuable addition to this work. I would also like to thank Dr. Mike Collins and Fred Pineau who facilitated my summer practicum assignments at Knolls Atomic Power Laboratory. The interactions and opportunities opened to me through the fellowship program have been one of the highlights of my time in graduate school.

I also wish to thank my other committee members, Dr. Jack Howell and Dr. Sheldon Landsberger. Dr. Landsberger first recruited me into the nuclear program at UT. I will always be indebted for the countless opportunities he has helped open to me.

I am blessed to have made some wonderful friendships within the nuclear and radiation engineering program and the mechanical engineering department at UT. Without their friendship and encouragement I might not have endured through my graduate studies. Finally, a sincere thank you is owed to my parents, Ray and Susan Reinert, and my family whose unrelenting support and inspiration has helped me achieve endeavors I might not have ever sought alone.

Investigation of Stochastic Radiation Transport Methods in Random Heterogeneous Mixtures

Publication No._____

Dustin Ray Reinert, Ph.D.

The University of Texas at Austin, 2008

Supervisors: Steven Biegalski and Erich Schneider

Among the most formidable challenges facing our world is the need for safe, clean, affordable energy sources. Growing concerns over global warming induced climate change and the rising costs of fossil fuels threaten conventional means of electricity production and are driving the current nuclear renaissance. One concept at the forefront of international development efforts is the High Temperature Gas-Cooled Reactor (HTGR). With numerous passive safety features and a meltdown-proof design capable of attaining high thermodynamic efficiencies for electricity generation as well as high temperatures useful for the burgeoning hydrogen economy, the HTGR is an extremely promising technology. Unfortunately, the fundamental understanding of neutron behavior within HTGR fuels lags far behind that of more conventional water-cooled reactors. HTGRs utilize a unique heterogeneous fuel element design consisting of thousands of tiny fissile fuel kernels randomly mixed with a non-fissile graphite matrix. Monte Carlo neutron transport simulations of the HTGR fuel element geometry in its full complexity are infeasible and this has motivated the development of more approximate

computational techniques. A series of MATLAB codes was written to perform Monte Carlo simulations within HTGR fuel pebbles to establish a comprehensive understanding of the parameters under which the accuracy of the approximate techniques diminishes. This research identified the accuracy of the chord length sampling method to be a function of the matrix scattering optical thickness, the kernel optical thickness, and the kernel packing density. Two new Monte Carlo methods designed to focus the computational effort upon the parameter conditions shown to contribute most strongly to the overall computational error were implemented and evaluated. An extended memory chord length sampling routine that recalls a neutron's prior material traversals was demonstrated to be effective in fixed source calculations containing densely packed, optically thick kernels. A hybrid continuous energy Monte Carlo algorithm that combines homogeneous and explicit geometry models according to the energy dependent optical thickness was also developed. This resonance switch approach exhibited a remarkably high degree of accuracy in performing criticality calculations. The versatility of this hybrid modeling approach makes it an attractive acceleration strategy for a vast array of Monte Carlo radiation transport applications.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction and Overview

Interest is rapidly growing in the design and deployment of the next generation of nuclear reactors needed to address rising domestic and worldwide energy demands. One concept that has garnered considerable interest in recent years is the High Temperature Gas-Cooled Reactor (HTGR). These types of reactors employ a fuel element design consisting of a random heterogeneous mixture of fissile fuel and non-fissile moderator.

Stochastic, or Monte Carlo, transport methods are inherently well suited to model the complex geometry of such a heterogeneous system in full detail. The Monte Carlo method operates by simulating the interactions of individual particles passing through the heterogeneous model. The expected value of some physical quantity of interest can then be approximated as the mean value of the individual simulations. A thorough analysis of the random heterogeneous media transport problems of interest in HTGR applications is arduous because of the hundreds of thousands of distinct spatial regions within such a reactor. In these immensely complex systems, it is extremely computationally expensive to simulate particle transport within a statistically accurate representation of a HTGR.

Given the colossal scope of a detailed model of such a system, recent research efforts have proposed novel methods for modeling the unique geometry of the heterogeneous fuel elements. These methods rely heavily upon simplifying assumptions to homogenize material cross sections or approximate or restrict the path lengths of particle travel between material interfaces. Such assumptions may yield significant errors when performing numerical simulations.

The principle goals of this research are:

1) Implement and evaluate Monte Carlo computational algorithms for calculating the transport of neutral particle radiation within spatially random heterogeneous materials.

2) Establish a more comprehensive understanding of how particular material and geometry configurations affect the suitability of the approximate computational techniques.

3) Develop and evaluate an optimized approach for particle transport in random heterogeneous media transport problems.

1.1 BACKGROUND

A random heterogeneous material is a mixture of two or more immiscible materials whose properties at any given spatial location are known only in a statistical sense (Davis, 2004). The problem of particle transport within random heterogeneous mixtures has received considerable attention over the past several decades in such fields as atmospheric modeling (Malvagi, 1992), two-phase coolant (Palmer, 2003) and groundwater flow through geologic materials (Lu, 2002). More recently, a considerable amount of attention among the nuclear reactor physics and radiation shielding community has focused upon the importance of understanding particle transport in random heterogeneous mixtures. This work has been motivated largely by the surge of interest in high temperature gas-cooled reactors (HTGRs).

The use of gas coolants in reactors is attractive because gases are typically easy to handle, have very low neutron interaction cross sections, can be operated at high temperatures without a phase change, and do not require pressurization (El-Wakil, 1982). Helium gas is the most commonly used coolant in HTGR designs. A central feature of

the HTGR is a very low power density and very large thermal capacity core such that the peak temperature reached under a worst-case complete loss of coolant scenario is substantially below the fuel melting temperature (Kadak, 2005).

The primary disadvantage of gas-cooled reactors is the relative inefficiency of gaseous heat transfer. In order to obtain a reasonable thermodynamic efficiency, gas-cooled reactors must be operated at very high temperatures. Unlike conventional water-cooled reactors, HTGRs are not restricted by coolant boiling. The maximum HTGR coolant outlet temperature is limited only by the material properties of the fuel and cladding. Because of their performance in high temperature applications, ceramic fuels were historically the major focus of fuel development for HTGRs (EL-Wakil, 1982).

HTGRs use a heterogeneous fuel type in the form of fissile particles dispersed in a continuous matrix of a non-fissile moderating material. The main advantage of so-called dispersion fuel elements over traditional homogeneous ceramic or metal fuel elements is an increased resistance to radiation damage (Samoilov, 1968). By utilizing individual localized regions of fuel, the resulting heterogeneous fuel-matrix mixture will be more stable under high irradiation conditions than will a comparable homogeneous fuel mixture. This is attributed to the localization of fission product accumulation. Some of the fission products produced within the fissile particles recoil into the adjacent matrix region. This recoil range is roughly 10 μm in most solid materials (Holden, 1966; Samoilov, 1968). Thus, it is desirable for the fuel element to have a mean distance between fissile particles of at least twice this fission product recoil range. Even under worst-case, high irradiation conditions, such a fuel element design would maintain a continuous web of undamaged matrix material.

For an array of spherical fissile particles of uniform diameter, the distance d between particles is given by

$$d = \left[\left(\frac{\pi}{3\sqrt{2}V_f} \right)^{1/3} - 1 \right] \cdot D \qquad (1.1)$$

where D is the fissile particle diameter, and V_f is the volume fraction of fissile particles (Holden, 1966). Thus, the fuel-matrix region should be designed such that this inter-particle spacing exceeds two times the recoil range of the fission fragments. Thus, a sparse packing of fissile particles is desired. This constraint will ensure a contiguous region of undamaged matrix. Figure 1.1 below illustrates how matrix damage changes with fissile particle size for a given fissile particle packing density assuming an average recoil range of 10 microns. Ideally, the fissile particles should be as small as possible to ensure an even radial fuel burnup. Figure 1.1 shows that the volume of the matrix region damaged by fission product recoil fragments increases rapidly for fissile particle diameters below 100 μm (Goeddel, 1962).

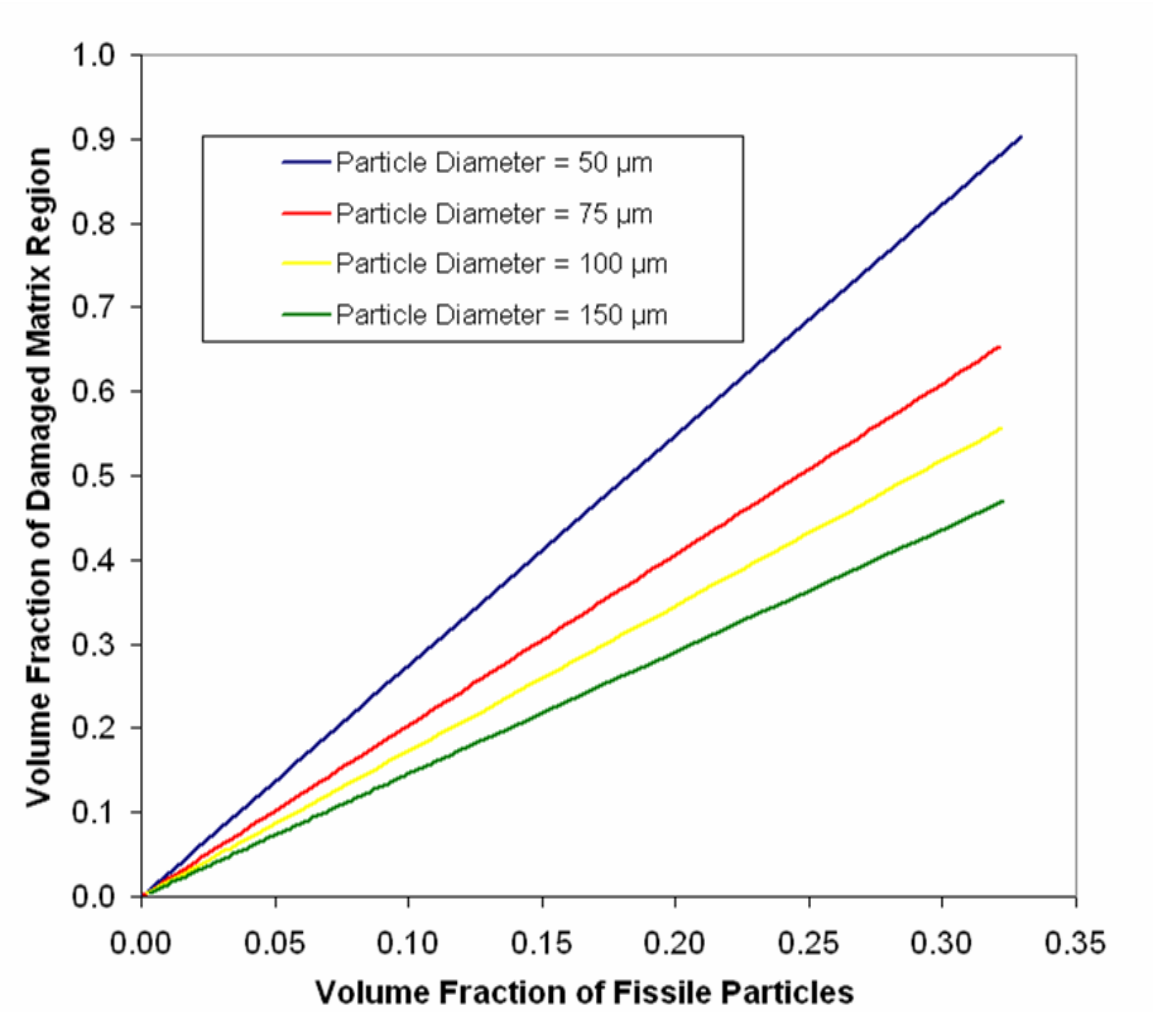


Figure 1.1: Variation of Matrix Damage Fraction with Particle Size and Volume Fraction (Goeddel, 1962)

One of the critical features of all contemporary HTGR designs is the containment of fission products directly at their source using a spherical fuel particle design known as a coated fuel particle (General Atomics, 2004). This design has been developed to allow for the high temperatures and high fuel burnup demanded of HTGR operation. The coated fuel particle is the primary barrier to fission product release. The fissile particles

are surrounded by coating layers that function as a barrier to fission product gas release. Because of their small size and simple, spherical shape, the coating layers can withstand the high pressures within the fuel particle caused by fission product gas buildup and irradiated fuel expansion.

Research begun in Germany in the 1960's led to the development of the first coated fuel particle design for use in the Thorium High Temperature Reactor (THTR) (Nickel, 2002). The coated fuel particle contained a central thorium-uranium dioxide kernel and a single pyrolitic carbon coating layer. More recently, two additional coated fuel particle designs have undergone significant development and testing: a two layer design known as BISO, with an inner porous carbon buffer region and outer dense pyrolitic carbon layer, and a four layer design, called TRISO (tri-structural isotropic), with a porous buffer region and a silicon carbide layer sandwiched between two denser pyrolitic carbon layers. The advantage of the multi-layer design is that fission products must cross multiple barriers before being released to the environment. TRISO particles have exhibited better retention of metallic fission products including cesium, strontium, and silver at high temperatures (Nickel, 2002). Low enriched uranium TRISO particles have been adopted for use in all HTGR projects currently under development (Tang, 2002).

Traditionally, the fissile kernels are formed using a gel droplet precipitation process. A uranium solution flows through a vibrating nozzle with a preset frequency. Spherical droplets are emitted and fall into an ammonia solution to form gel particles. After calcination and sintering, the kernels are sorted to ensure uniformity in size and shape. The carbide coating layers are applied using chemical vapor deposition in a fluidized bed furnace purged with argon (Tang, 2002).

The reactor fuel elements are constructed by immersing the coated fuel particles in a solid graphite matrix. The coated fuel particles are mixed with graphite flour and compacted under high pressures to form fuel elements. In addition to its excellent neutron moderating capability, the high surface to volume ratio of the coated particles and the high heat conductivity of the graphite matrix promotes heat transfer (Bende, 1999). Efficient heat transfer allows for very high coolant exit temperatures and thus higher thermodynamic efficiencies. The graphite matrix must also retain its strength at high temperatures, resist thermal and radiation induced stresses as well as corrosion brought about by potential impurities in the gas coolant (Holden, 1966).

This coated fuel particle and graphite matrix fuel design affords a great deal of flexibility of application. The enrichment, effectively the moderator to fuel ratio, can be varied simply by altering the number of coated fuel particles within the fuel region. Fuel particles representative of thorium and plutonium fuel cycles have also been successfully demonstrated (Nickel, 2002). The very high temperatures also make the HTGRs an ideal system for hydrogen production (Koberl, 2004).

There are two distinct classes of HTGRs that use spherical coated fuel particles. The first is the prismatic core HTGR that has undergone development in the United States, the United Kingdom, and most recently in Japan (Nickel, 2002). The fuel element consists of hexagonal shaped graphite blocks approximately 1 m long and 0.4 m across. Each hexagonal assembly has fuel, coolant, and control rod channels in a hexagonal array. Cylinder shaped fuel compacts 80 mm in length and 26 mm in diameter are stacked in the fuel channels. Each compact contains coated fuel particles dispersed in graphite matrix (Kunitomi, 2004).

The second fuel design is a spherical fuel element first utilized in the German Arbeitsgemeinschaft Versuchsreaktor (AVR) in the 1960's. This type of reactor design is

generally referred to as a pebble bed reactor. In the Japanese-designed HTR-10 modular HTGR the coated fuel particle and graphite matrix mixture is pressed into a 60 mm spherical pebble consisting of a 5 mm thick outer graphite outer shell and a central fuel-matrix region. Each pebble contains approximately 8,300 coated fuel particles. A total of 27,000 spherical fuel pebbles are loaded in the reactor core. Reactivity control is achieved by the addition or removal of fuel pebbles from the core hopper (Tang, 2002). A representative TRISO coated fuel particle and fuel pebble is depicted in Figure 1.2.

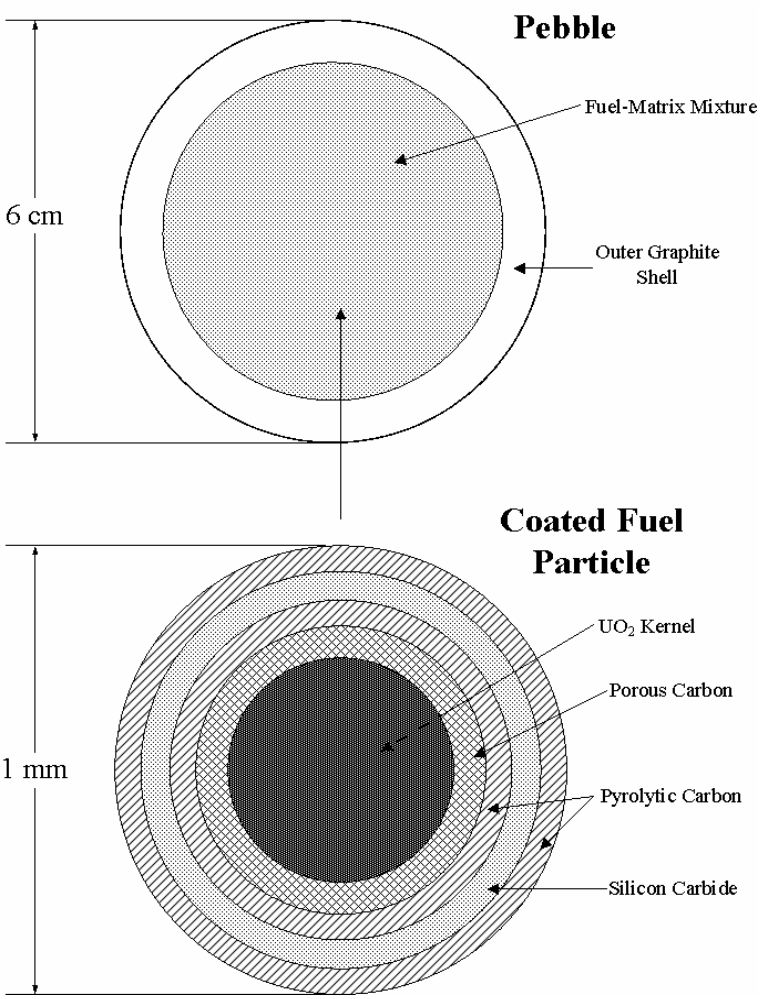


Figure 1.2: TRISO Coated Fuel Particle and Pebble Arrangement

Both the prismatic and pebble bed reactors have two layers of system heterogeneity caused by the random arrangements of the spherical coated fuel particles within the randomly located cylindrical or spherical fuel elements (Murata, 1996). Because of the random spatial heterogeneity of the both the coated fuel particles and the fuel pebbles, the precise locations of the fuel and moderator materials are unknown. The heterogeneity also makes it challenging to introduce effective cross sections of the fuel regions into conventional radiation transport codes (Murata, 1996). In theory, the heterogeneity effects could be obviated by repeating a large number of calculations each having a specific realization of a random geometrical arrangement. However, commercial pebble bed reactors may contain 10^9 to 10^{10} total coated fuel particles. The immense scope of a system having billions of discrete spatial regions renders this type of brute force method of computation infeasible.

More approximate, less computationally burdensome methodologies have been developed to analyze systems such as HTGRs for nuclear engineering applications. These approximate methods are the central focus of this thesis. With a more thorough understanding of these methodologies and the applicability of their use, we can hope to attain more accurate results with less computational expense and also garner additional information about how the random material distributions affect neutron transport within these mixtures.

1.2 DISSERTATION ORGANIZATION

The purpose of this dissertation is to extend the understanding of existing, traditional computational methodologies and also propose new, novel stochastic transport methods for performing neutron transport calculations in random heterogeneous media.

The organization of this dissertation and a summary of the contributions of this research are described below.

Chapter 2 gives an overview of neutron transport in random heterogeneous mixtures with the introduction of the governing mathematical equations and the Monte Carlo method of solution. Existing computational methodologies are explained and the relative advantages and disadvantages of each method are discussed. Chapter 3 describes the development of a Monte Carlo analysis code for performing neutron transport simulations in binary (two-phase) random media. Chapter 4 describes the assessment of dimensionless parameters for describing the geometric and material properties of a given random heterogeneous mixture. The ability of this parameter set to predict the accuracy of a particular solution method is also evaluated. The accuracy of the chord length sampling technique was discovered to be a function of the total optical thicknesses and optical scattering thickness of the constituent materials as well as the packing density of the fissile kernels. The results of this parameter assessment provide a foundation for the original algorithm development described in chapters 5 and 6. Chapter 5 discusses the implementation of an extended memory chord length sampling routine. The extended memory chord length sampling routine recalls a neutron's prior material traversals. This research validates the effectiveness of this technique in performing fixed source, three-dimensional transport simulations of densely-packed, optically thick kernels. Chapter 6 narrates the development of a new hybrid algorithm that combines homogeneous and explicit geometry models within a single Monte Carlo simulation. The geometry model utilized is selected according to the energy dependent optical thickness. The initial success of this hybrid modeling approach makes it an attractive methodology for future application to a wide assortment of radiation transport analyses. Conclusions from this research and recommendations for future work in this field are presented in Chapter 7.

Chapter 2: Theory

2.1 THE NEUTRON TRANSPORT EQUATION

The starting point for almost all problems of interest in the field of numerical radiation transport is the neutron transport equation. This mathematical formulation is derived from the Boltzmann equation of gas dynamics. Neutrons are assumed to propagate independently of one another, following linear paths between collisions. The conventional transport equation in the context of nuclear engineering is a linearized form of the Boltzmann equation based upon particle conservation within a finite control volume.

Equation 2.1 describes the space, energy, and time distribution of neutrons within a given phase-space region as depicted in Figure 2.1 below.

$$\frac{1}{v}\frac{\partial\psi\left(\hat{r},E,\hat{\Omega},t\right)}{\partial t}+\hat{\Omega}\cdot\nabla\psi\left(\hat{r},E,\hat{\Omega},t\right)+\sum_t\left(\hat{r},E,t\right)\cdot\psi\left(\hat{r},E,\hat{\Omega},t\right)= \\ S\left(\hat{r},E,\hat{\Omega},t\right)+\int_0^{\infty}dE'\int_0^{4\pi}d\Omega'\,\sum_s\left(\hat{r},E\rightarrow E',\hat{\Omega}'\rightarrow\hat{\Omega},t\right)\psi\left(\hat{r},E',\hat{\Omega}',t\right) \tag{2.1}$$

The terms of the left hand side of equation 2.1 denote the time rate of change of the neutron population within the region and the loss of neutrons attributed either to leakage from the volume or collision losses that yield absorptions or scattering interactions that result in the neutron's energy or direction being outside the region of interest. The right hand side of the balance equation denotes source terms including external and fission sources as well as in-scattering of neutrons from a different energy or direction interval. The principal quantity of interest in the above equation is the angular

flux, $\psi(\hat{r}, E, \hat{\Omega}, t)$, which describes the density of neutrons at position $\hat{r}$, with energy E, moving in direction $\hat{\Omega}$, at time t.

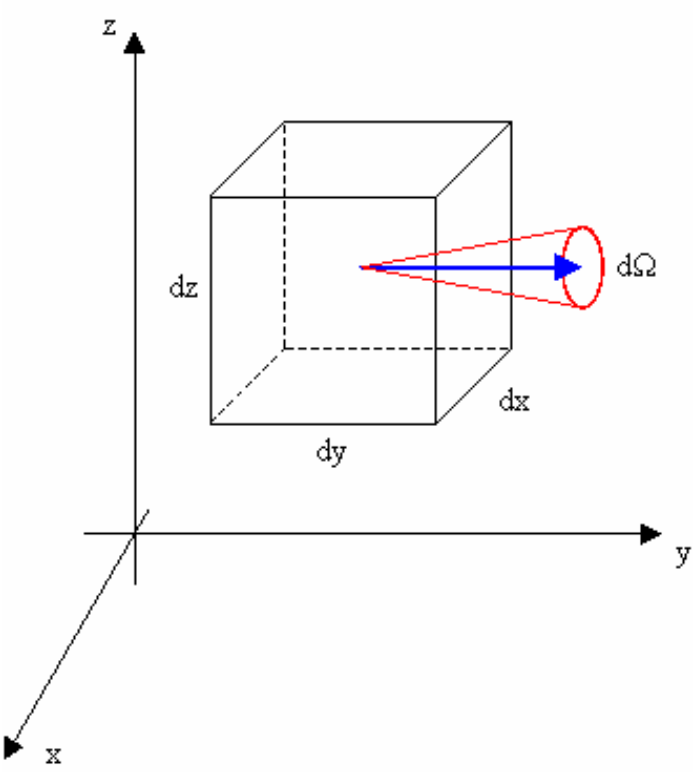


Figure 2.1: Infinitesimal phase-space volume element considered by the quantity $\psi(\hat{r}, E, \hat{\Omega}, t)$ used in the Neutron Transport Equation. $\psi(\hat{r}, E, \hat{\Omega}, t)$ describes neutrons that have an energy dE about E, within time interval dt, moving within a solid angle of $d\Omega$ about $\hat{\Omega}$ (Leppanen, 2007).

If one considers only the steady-state neutron population within the phase-space region, equation 2.1 above simplifies to

$$\hat{\Omega} \cdot \nabla \psi(\hat{r}, E, \hat{\Omega}) + \Sigma_t(\hat{r}, E) \cdot \psi(\hat{r}, E, \hat{\Omega}) = S(\hat{r}, E, \hat{\Omega}) + \int_0^{\infty} dE' \int_0^{4\pi} d\Omega' \, \Sigma_s(\hat{r}, E \to E', \hat{\Omega}' \to \hat{\Omega}) \psi(\hat{r}, E', \hat{\Omega}') \tag{2.2}$$

The steady-state form of the transport equation given above has a total of six independent variables: 3 variables in space (x, y, and z), 2 variables in angle (θ and ϕ), and a single variable in energy, E. Thus, the above expression is still quite difficult to solve analytically except for the most simplistic applications. Therefore, the transport equation is almost always solved with numerical techniques.

In deterministic transport methods, the neutron transport equation is converted into a system of discretized equations that can be readily solved using a computer. Most commonly, the energy domain is divided into discrete groups, the spatial domain is mapped to a spatial mesh, and the angular variable is represented along a finite set of discrete directions. Material properties are assumed uniform in each spatial element. The discretized system of equations is then solved for the angular flux in each discrete phase-space element and yields an approximate solution to the neutron transport. Aside from possible uncertainties in the material cross section data, the only source of error in this class of techniques is the systematic computing error arising from the discretization of each of the independent variables. However, because of their immensely complex geometries, deterministic methods are not well suited to model HTGR fuels.

2.2 MONTE CARLO TRANSPORT FUNDAMENTALS

Stochastic transport methods take advantage of the fact that the passage of radiation through matter is an inherently stochastic process. The fate of a given neutron is determined by a succession of random neutron-material interactions. Stochastic, or Monte Carlo methods, simulate these physical interactions. In stark contrast to deterministic techniques which yield a direct solution, Monte Carlo results are inferred as the average observed behavior of a large set of stochastic numerical simulations. However, Monte Carlo methods do not directly invoke the neutron transport equation or

any other governing physical equations. All that is needed for a Monte Carlo calculation is a thorough understanding of the stochastic processes that affect neutron interactions. Each stochastic process is mapped to a probability distribution function. Random number generators are used to sample from probability distributions that describe neutron scattering angles, post-collision energies, secondary particle production, and free-flight path lengths between collisions.

Each individual neutron simulation, called a history, begins by sampling a source distribution to assign the neutron's initial position, energy, and direction. The distance to the first interaction site, the isotope with which the collisions occurs, and the type of reaction are then randomly sampled according to probability distribution functions defined by the material properties. If the interaction results in absorption, the neutron's history is terminated, otherwise the random walk tracking process continues until the neutron is absorbed or leaks from the system.

Physical quantities of interest can be determined by tallying and averaging the observed behavior of the individual histories. At least one region of interest, known as a tally region, must be defined in each Monte Carlo calculation. Events that occur within this region are counted. Events of interest may include a neutron history traversing a particular tally region or collisions resulting in a capture, fission, or other reaction. Results are obtained by assigning a score, x_i, to each neutron history, where i denotes the i^{th} random walk simulation. x_i can range from 0 to 1. If the neutron never reaches the tally region then x_i is 0. Conversely, if the neutron reaches the tally region with little or no prior interactions the score is very near 1. The probability that a given history will score between x and $x + dx$ is denoted as $p(x)dx$ where $p(x)$ is a probability distribution function (Shultis and Faw, 2006). The expected or true mean of such a distribution is then

$$\langle x \rangle = \int_0^\infty x p(x) dx. \tag{2.3}$$

Unfortunately, the actual distribution is rarely known. The ability to approximate this distribution lies at the heart of the Monte Carlo technique. The true mean is approximated as the sample mean determined by the Monte Carlo simulation. This sample mean is simply

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{2.4}$$

where x_i is the tally contribution from the ith neutron history and N is the total number of histories. The results of a Monte Carlo calculation are averages of tally contributions from many neutron histories. Because the sample mean is the result of a stochastic computational experiment, it will have some associated uncertainty. The standard deviation of scores, x_i, sampled during a Monte Carlo simulation is given by

$$S^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(x_i - \bar{x} \right)^2 . \tag{2.5}$$

which for large N

$$S^2 \cong \overline{x^2} - \bar{x}^2 \tag{2.6}$$

where

$$\overline{x^2} \equiv \frac{1}{N} \sum_{i=1}^{N} x_i^2 . \tag{2.7}$$

The estimated variance of the sample mean is then

$$S_{\bar{x}}^2 = \frac{1}{N} S^2 \quad \text{(Shultis and Faw, 2006).} \tag{2.8}$$

Such results can be obtained for both fixed source, or shielding, and criticality, or fissioning medium, calculations.

The disadvantage of the Monte Carlo method is that it doesn't produce an exact solution. All results are purely observed estimates and include a statistical uncertainty.

Monte Carlo calculations can be very time consuming if very high precision is desired or if the probability of obtaining a non-zero tally contribution, x_i, is small. While the Monte Carlo method affords great deal of flexibility in modeling an exact representation of a problem, it can only yield an approximate solution (Mendius, 1994).

2.3 GEOMETRY TRACKING

A major advantage of the Monte Carlo method is that the geometry and material properties of the problem may be modeled without discretization. The Monte Carlo method can deal with almost any complex arbitrary geometry. Deterministic methods must spatially discretize the problem space into a series of polygonal cells. As such, deterministic techniques are limited to reasonably simple geometries. With Monte Carlo methods, the truncation error arising from this spatial discretization can be entirely avoided; thus complex three-dimensional configurations may be modeled with very high fidelity. The building blocks of a Monte Carlo geometry model are primitive geometrical bodies and polynomial surface definitions. Modern surface-based Monte Carlo codes can accommodate arbitrary linear, quadratic, and quartic surfaces (Hughes, 2007). This enables the modeling of almost completely arbitrary geometries. This flexibility is often required to appropriately model the detailed heterogeneous material distributions present in reactor physics or radiation shielding applications (Brown, 2004b).

The problem of determining the distance to the next geometry interface and the distance the next collision is the most frequent calculation performed in every Monte Carlo simulation. This distance determination has two parts. Given a specific neutron history at a particular spatial location and traveling a particular direction:

1) calculate the distance to the nearest geometry interface

2) sample, using the appropriate total cross section, the distance to the next collision.

The minimum of these two distances determines the next event in the neutron's history. A standard reactor simulation involves on the order of 10^{10} distance calculations (Brown, 2005). Thus, it is imperative that this part of the simulation is performed in an efficient manner.

With conventional techniques, the only way to determine the distance to the nearest surface is to calculate the distance to all surfaces along the current direction, or "line-of sight", and then select the shortest of these distances. This can be a very time consuming calculation if the mean path length is long in comparison to the spacing between surfaces. This distance to nearest surface calculation must be performed after every surface crossing. If the neutron traverses many surfaces without interacting, this calculation will have to be performed many times between successive collisions (Leppanen, 2007).

This can also be particularly problematic if there are many distinct surfaces within a particular computational region. This is illustrated in the cross section of the cylindrical fuel assembly depicted in Figure 2.2 below. To track a neutron within region A, a total of nine surfaces would have to be considered for every distance to nearest surface determination: the inner and outer boundaries as well as all eight individual fuel pins. A much more efficient calculation would be to subdivide region A into individual sub-regions so that only those surfaces residing within a particular sub-region were considered for each distance to nearest surface calculation (Brown, 2003).

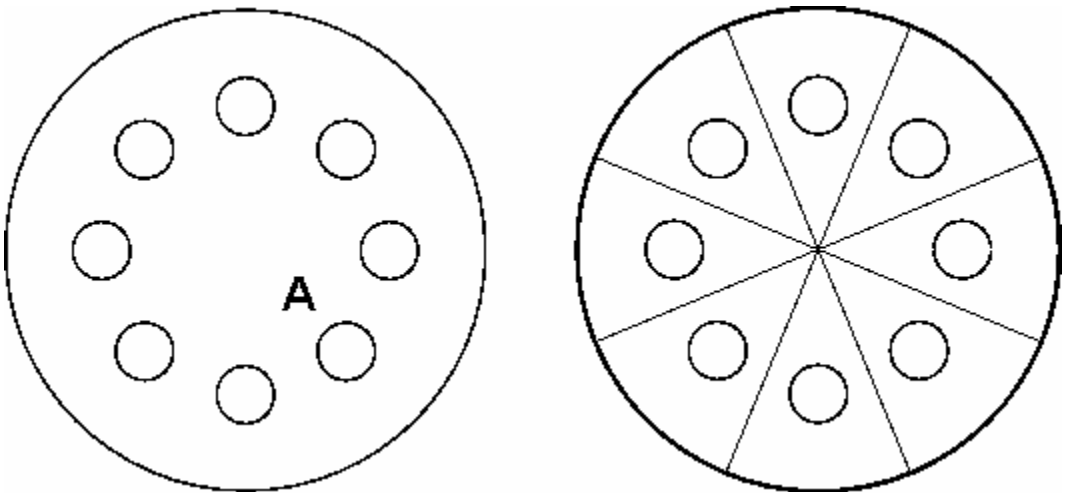


Figure 2.2: Geometry Binning for Improved Monte Carlo Simulation Efficiency (Brown, 2003)

This partitioning technique is less straightforward if the locations of all surfaces are not rigidly defined and regularly spaced. Thus, in the analysis of a HTGR fuel assembly, affixing cell boundaries to the randomly dispersed coated fuel particles and still preserving the true random nature of the mixture significantly complicates the analysis procedure.

2.4 NEUTRON TRANSPORT IN RANDOM MEDIA

The remainder of this thesis focuses upon Monte Carlo transport methods for neutron transport in heterogeneous mixtures consisting of two or more randomly mixed immiscible materials. That is, the material present at any given point in the mixture is known only according to some presumed mixing statistics. Physically, this can be thought of as a grainy material consisting of randomly distributed chunks of two or more distinct components (Pomraning, 1998).

There has been a great amount of effort over the past several decades devoted to the development of stochastic models for performing transport calculations in such materials. These models are essentially modifications of the neutron transport equation aimed to better describe the statistical nature of the random heterogeneous medium. Below we introduce some of the most significant such models and their application in Monte Carlo transport simulations.

2.4.1 Benchmark Method

If a complete physical realization of a random heterogeneous material is defined explicitly using the nominal mixing statistics, a solution for the angular flux in that particular realization can be found using conventional Monte Carlo techniques. This calculated solution, however, represents only one of a possibly infinite number of physical realizations of the presumed known statistics of the random mixture. A brute force approximation could be attained by creating many physical realizations of the random heterogeneous material, solving each for the particular quantity of interest, and then taking the average over all realizations. This is the technique employed by Adams *et al*. (1989) and is often referred to yield the "benchmark" ensemble-averaged flux solution. This method requires no essential approximations and given an infinite number of random realizations and an infinite number of neutron histories the benchmarking process will produce a true ensemble average, or mean, solution for the angular flux. Although intuitive in application, the overwhelming disadvantage of the benchmark method is the computational expense required of problems of any practical significance. Consequently, most research efforts have focused on the development of methods which approximate these "benchmark" results while expending less computational effort. The only practical usefulness of the benchmark model has proven to be in verifying the validity of these other more approximate methods (Miller, 2000).

2.4.2 Atomic Mix Model

The simplest and most widely used method of approximation is known as the atomic mix model. With this method, the volume fraction of each material within the random heterogeneous mixture is used to calculate a homogenized cross section. The homogenized cross section for a mixture of two immiscible materials is determined according to

$$\overline{\Sigma} = p_0\Sigma_0 + p_1\Sigma_1 \tag{2.9}$$

where p_0 and p_1 define the probabilities of finding each material at any given point within the immiscible mixture. With this method, the distinct spatially-dependent material properties in the neutron transport equation are replaced by their volume weighted averages.

For example, consider the steady-state, energy-independent neutron transport equation

$$\hat{\Omega}\cdot\nabla\psi\left(\hat{r},\hat{\Omega}\right)+\Sigma_t\left(\hat{r}\right)\cdot\psi\left(\hat{r},\hat{\Omega}\right)=S\left(\hat{r},\hat{\Omega}\right)+\int_0^{4\pi}d\Omega'\Sigma_S\left(\hat{r},\hat{\Omega}'\rightarrow\hat{\Omega}\right)\psi\left(\hat{r},\hat{\Omega}\right) \tag{2.10}$$

The material properties appearing above are $\Sigma_t(\hat{r})$, $\Sigma\left(\hat{r},\hat{\Omega}'\rightarrow\hat{\Omega}\right)$ and $S\left(\hat{r},\hat{\Omega}\right)$ Thus, the ensemble average angular flux, $\left\langle\psi\left(\hat{r},\hat{\Omega}\right)\right\rangle$, can be found from only a single homogeneous calculation.

Formal transport equations implementing the atomic mix approximation for a variety of cases have been previously described by several authors (Donovan, 2003c; Miller, 2000; Pomraning 1991) and their methods are reiterated here.

All random quantities appearing in the neutron transport equation can be expressed as the sum of an ensemble average, or mean, component and a fluctuating component where by definition the ensemble average of this fluctuating term is exactly zero.

$$\psi = \langle \psi \rangle + \tilde{\psi} \tag{2.11}$$

$$\Sigma = \langle \Sigma \rangle + \tilde{\Sigma} \tag{2.12}$$

$$\langle \Sigma \psi \rangle = \langle \Sigma \psi \rangle + \langle \tilde{\Sigma} \tilde{\psi} \rangle \tag{2.13}$$

Substituting these values into equation 2.10 gives, for any localized region,

$$\begin{aligned} &\hat{\Omega} \cdot \nabla \langle \psi(\hat{r}, \hat{\Omega}) \rangle + \langle \Sigma_t(\hat{r}) \cdot \psi(\hat{r}, \hat{\Omega}) \rangle + \langle \tilde{\Sigma}_t(\hat{r}) \cdot \tilde{\psi}(\hat{r}, \hat{\Omega}) \rangle = \\ &\langle S(\hat{r}, \hat{\Omega}) \rangle + \int_0^{4\pi} d\Omega' \langle \Sigma_S(\hat{r}, \hat{\Omega}' \to \hat{\Omega}) \psi(\hat{r}, \hat{\Omega}) \rangle + \langle \tilde{\Sigma}_S(\hat{r}, \hat{\Omega}' \to \hat{\Omega}) \tilde{\psi}(\hat{r}, \hat{\Omega}) \rangle \end{aligned} \tag{2.14}$$

The problem is that the above equation no longer contains just a single unknown. The ensemble average flux $\langle \psi \rangle$ appears along with multiple cross correlation terms of the form $\langle \tilde{\Sigma} \tilde{\psi} \rangle$. The central assumption of the atomic mix approximation is that all cross correlation terms $\langle \tilde{\Sigma} \tilde{\psi} \rangle$ are negligible. Physically, this is equivalent to assuming that at any local region within the mixture, the material properties are sufficiently represented by the ensemble average values.

The advantage of the atomic mix model for transport random heterogeneous mixtures is readily apparent: a single realization defined by only one unique spatial region can be used to predict an ensemble average solution for the mixture. This approximation does yield acceptable results in many situations. However, the atomic mix model has been shown to be accurate only if all material chunks are optically thin (Shultis and Faw, 2006; Torquato, 2002). That is,

$$\Sigma_i \times \lambda_i << 1 \tag{2.15}$$

where λ_i is the chord length, or average free flight path, of a neutron within material i. If the optical thickness of all constituent regions of a heterogeneous material is very small, then there is a high probability that a neutron will be transmitted through many regions between successive collisions. Thus, homogenizing the material into a single continuous mixture may not significantly obscure results. In contrast, the atomic mix model can also

yield highly erroneous results if applied without discretion. Large errors result, for instance, if the approximation is applied to problems involving a highly localized, heavy absorber. The heavy absorption contribution is artificially spread throughout all of the computational model space and will overestimate the absorption effectiveness (Shultis and Faw, 2006).

2.4.3 Effective Homogenization

Effective homogenization can be thought of as an extension of the atomic mix method. The objective of this method is to not only approximate the interaction probabilities in terms of volume-averaged cross sections, but to also capture some of the effect of the size and dispersion of the interstitial heterogeneities.

Much of the early advancement in the area of effective homogenization theory was encompassed by the work of Doub (1961). Doub showed that for a non-scattering medium consisting of small poison spheres within a plate of homogenous matrix material, the transmission probability of a neutron through the plate could be expressed as the product of a particle self-shielding factor, f_0, and the transmission probability of an equivalent volume-averaged (atomic mix) homogeneous mixture. That is,

$$T\left(\Sigma_0, \Sigma_{matrix,} l\right) = e^{-lf_0\left[(1-V)\Sigma_{matrix}+V\Sigma_0\right]} \quad \text{(Donovan, 2003c)} \tag{2.16}$$

where Σ_0 and Σ_{matrix} denote the macroscopic cross sections of the spherical poison particles and matrix material, respectively, l is the average particle path length through the plate, V is the volume fraction of spherical particles in the plate and f_0 is the particle self-shielding factor. f_0 is the fraction of the original poison sphere material that, when homogenized, will yield the same transmission as the original mixture.

The key task at this point in effective homogenization method is to derive an expression for f_0. This always requires some type of simplifying assumption to characterize the random mixture. Doub's assumption was that the matrix material could

be represented as an equivalent system of densely packed spheres of radius equal to that of the poison spheres. This permitted a closed-form expression for the self-shielding factor which was a function of the ratio of poison spheres to matrix spheres, and the averaged transmission probability through a single representative stochastic sphere

More recent work in effective homogenization for random media (Morel, 2003; Yamamoto, 2006) has used similar approaches. Yamamoto's approximation of coated fuel particles divided the random mixture into thin layers and assumed that a transported particle would interact with no more than one coated fuel particle per layer. The effective cross section was then formulated by a detailed integration over the particle's flight path.

Effective homogenization has the potential to produce results with greater accuracy than those obtained with the atomic mix method. However, this technique can be plagued by the often significant assumptions necessary to express the self-shielding factor analytically and the increased computational expense required to numerically approximate this factor for more complicated geometries.

2.4.4 Levermore-Pomraning Model

Early work in the formulation of a general stochastic transport equation for accurately describing particle transport in random heterogeneous mixtures was investigated using a Liouville master equation approach (Vanderhaegen, 1986). The Liouville master equation is generally applied to initial value problems. Vanderhaegen showed that this technique could also be applied to steady-state, non-scattering particle transport problems with Markovian mixing statistics, with the spatial coordinate replacing the time coordinate as the independent variable (Pomraning, 1991). Markovian mixing statistics means that within a mixture, the probability of transitioning from material i to material $j \neq i$ in a given short distance ds is ds/λ_i. This corresponds to a mean free-flight path across material i of λ_i. This is the usual assumption made in

modeling stochastic mixtures. The λ's are also known as the Markovian transition lengths or mean chord lengths of each material (Olson *et al*., 2003). Thus, in a binary mixture consisting of two components i and j, λ_i and λ_j are sufficient to completely define the volumetric mixing statistics of the mixture (Donovan, 2003). That is,

$$p_i = \lambda_i / (\lambda_j + \lambda_i) \tag{2.17}$$

$$p_j = 1 - p_i \tag{2.18}$$

where p_i and p_j are the volume fractions of each constituent within the binary mixture.

Further advancement of this work yielded a more useful general model that considered both time dependence and scattering (Pomraning, 1998). These results were matched by Adams *et al*., (1989), who approached the same problem from a particle conservation perspective. Adams *et al*. considered a mixture of two materials each having constant materials properties and separated by a boundary surface Γ. This is illustrated in Figure 2.3 below. A statement of particle conservation was then written for each of the two material regions.

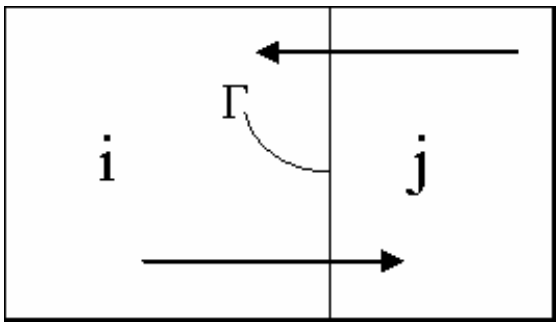


Figure 2.3: Particle Conservation for a Binary Mixture with Interface Coupling

The product of this work is a set of coupled conservation equations known as the general stochastic transport equation.

$$\hat{\Omega}\cdot\nabla p_i\psi_i + \Sigma_{t,i}\cdot p_i\psi_i = p_i S_i + \int_0^{\infty} dE' \int_0^{4\pi} d\Omega' p_i \Sigma_{s,i}\psi + \frac{p_j\overline{\psi}_j}{\lambda_j} - \frac{p_i\overline{\psi}_i}{\lambda_i} \quad , \quad j \neq i \tag{2.19}$$

where the indices $\left(\hat{r}, E, \hat{\Omega}\right)$ have been omitted for brevity. p_i is the probability that point $\hat{r}$ exists within material i, λ_i is the Markovian transition length from material i to material j, and ψ_i and ψ_j are ensemble average fluxes in each of the two materials. This equation includes two additional terms $\overline{\psi}_i$ and $\overline{\psi}_j$ that are present if the position $\hat{r}$ lies on the interface between the two materials. These are interface coupling terms that denote a flow of particles from material i to material j and from material j to material i respectively. This expression is formally exact and contains no essential approximations. However, with the inclusion of the interface coupling terms, the set of above equations has a total of four variables $\left(\psi_i, \psi_j, \overline{\psi}_i, \overline{\psi}_j\right)$ described by only two equations. Thus, an approximation is needed in order to simplify this system of equations into a useful form.

The simplest closure relation is known as the Levermore-Pomraning approximation.

$$\overline{\psi}_i = \psi_i \tag{2.20}$$

This assumption means that the flux at the interface is equal to the flux away from the interface within the bulk of the material. This closure relation is equivalent to the commonly used substitution for upwind differencing in numerical finite difference solutions to hyperbolic equations (Pomraning, 1998). Thus, the only remaining unknowns in Equation 2.19 are then ψ_i and ψ_j allowing a direct solution by standard deterministic transport techniques.

The Levermore-Pomraning model is exact only when applied to purely absorbing materials. For purely absorbing media, the flux at any particular point $\hat{r}$ is a function of

only the optical distance to $\hat{r}$ from all source points. For Markovian mixing statistics, this optical distance is independent of whether point $\hat{r}$ lies directly on an interface or within the bulk of one of the materials. When scattering is introduced, however, the Levermore-Pomraning closure relation becomes less accurate (Adams *et al*., 1989).

2.4.5 Chord Length Sampling

Shortly after the publication of the general stochastic transport equation, Sahni (1989) showed that applying the Levermore-Pomraning model with Markovian mixing statistics was equivalent to assuming that a particle's successive free-flight tracks were uncorrelated. In other words, each segment of a particle's history is independent of all previous segments of the history. This discovery led to the development of a Monte Carlo technique that has become known as chord length sampling.

Zimmerman and Adams (1991) suggested the use of probability distribution functions to describe the locations of the material interfaces within a random heterogeneous mixture. They proposed that if the distribution of chord lengths could be described for each material in terms of probability distribution functions, these functions could be incorporated into a Monte Carlo particle tracking algorithm. By sampling from the chord length distribution function, material interfaces are placed "on-the-fly" during the random walk process. Following every collision, the distance to the next material interface is sampled from the probability distribution functions rather than being calculated explicitly. This greatly reduces the computational expense of rigorously defining and tracking the particle's path through an entire explicit geometry realization.

The central challenge of chord length sampling lies in devising suitable chord length probability distribution functions from which to sample the "on-the-fly" placement of interfaces within the heterogeneous mixture. At some arbitrary position within

material i the probability of finding a chord of length λ within the random mixture is $p_i(\lambda)$. The probability of finding a chord of length between λ and $\lambda + d\lambda$ is

$$\int_0^\infty p_i(\lambda) d\lambda = 1 \text{ (Torquato, 2002).} \tag{2.21}$$

The characteristic Markovian transition length within material i, also known as the mean chord length, is the first moment of the chord length probability distribution function

$$\overline{\lambda} \equiv \int_0^\infty \lambda \cdot p_i(\lambda) d\lambda . \tag{2.22}$$

Thus $1/\overline{\lambda}$ is the probability of encountering an interface per unit path length of particle flight within material i. Further details regarding the determination of this mean chord length are discussed in Chapter 3.

The chord length sampling process is very similar to the conventional Monte Carlo technique of sampling the distance that a particle moves between collisions. Therefore, chord length sampling can be integrated into a Monte Carlo code with little accommodation. The chord length sampling algorithm suggested by Zimmerman and Adams is as follows. For a particle with a given position and direction,

1) calculate the distance to the problem boundary

2) sample the distance to a collision using the material's total cross section

3) sample the distance to material interface using the chord length distribution function

The minimum of these three distances determines the next event. If 1) the particle leaks from the problem boundary. If 2) the particle's position is updated, the appropriate collision physics is performed, and the distance determinations are repeated. If 3) the particle's position is updated, the material properties are changed, and the distance determinations are repeated. The history proceeds until the particle leaks from the system

or is absorbed. As stated previously, the bulk of the computation expense of the benchmark solution for HTGR fuels lies in the distance to next interface calculation. After every matrix collision, the benchmark model requires calculating the distance to thousands of coated fuel particles. In chord length sampling, the distance to next interface is sampled from a probability distribution function using a single random number.

Zimmerman and Adams (1991) demonstrated their technique for one-dimensional rod and slab geometries with good agreement with benchmark results. More recently, Murata *et al*. (1996) and Donovan (2003b) have applied the chord length sampling method in three dimensions. Both have demonstrated results that were far more accurate than those produced by the atomic mix method with only slightly greater computing expense.

The drawback of the chord length sampling method is that it solves an approximate set of transport equations. Hence, the assumptions brought forth by the introduction of the Levermore-Pomraning closure relation also plague the chord length sampling method. Chord length sampling, too, degrades in scattering media. The uncorrelated segments assumption is completely valid only in a purely absorbing medium in which there is no possibility of backscattering. If chord length sampling is applied in a highly scattering medium, there exists the possibility that a particle would encounter a particular material at a given location and then at a later time in its history scatter to find a different material at that same location. This is illustrated in Figure 2.4 below.

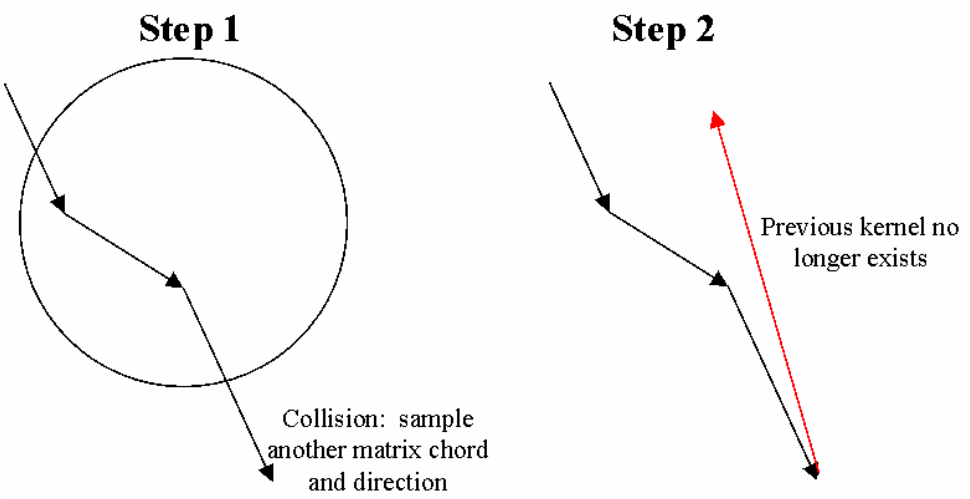


Figure 2.4: Geometry Error due to Backscattering Matrix Collision in Chord Length Sampling

The effects of such conflicting geometry errors are the subject of further discussion in Chapter 4.

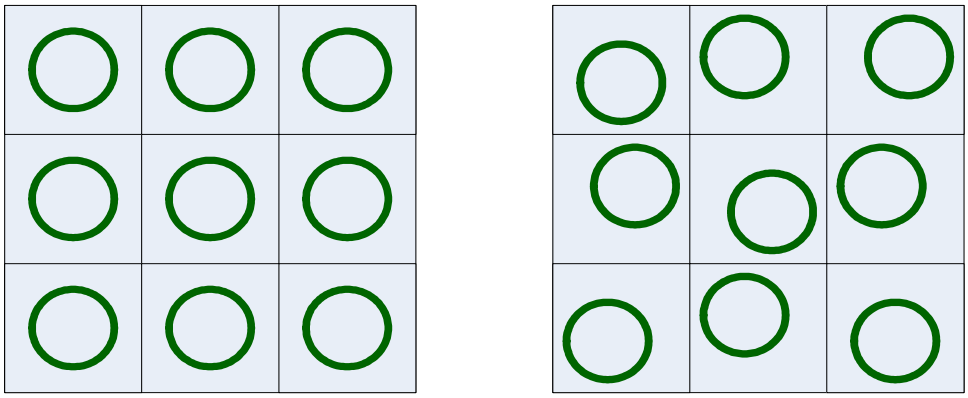
2.4.6 Lattice Methods

One straightforward method of modeling HTGR fuels is to affix the coated fuel particles to a lattice. The locations of the uranium particles are placed at regularly spaced locations within the graphite mixture. The characteristic dimension of the lattice is sized to preserve the nominal volumetric mixing statistics. All geometric regions are modeled explicitly. However, by negating the randomness of the mixture, the computational expense of particle tracking procedure can be greatly reduced.

The internationally recognized Monte Carlo N-Particle radiation transport code (MCNP) uses a lattice approach, modeling repeated structures using its "universe" concept (Mendius, 1994). A universe is a collection of either regular or arbitrary cells that are embedded inside of another cell, called a universe cell. Universe cells can then be embedded in another cell. One method to model a HTGR fuel pebble is to embed a

universe comprised of cells defining a single coated fuel particle and the surrounding graphite matrix material inside of a three-dimensional lattice. Thus, with the MCNP technique, the problem of rigorously constructing an explicit model of a HTGR fuel pebble can be approximated by defining a single regular lattice and a single representative coated fuel particle cell. Computation time is reduced significantly because the particle tracking algorithm has only to consider path lengths within the current cell and the distances to the few neighboring lattice cells. Conversely, a benchmark algorithm with explicit geometry definitions must repeatedly consider path lengths associated with each of the many thousand of cells.

Fixing the fuel kernel locations to a lattice, however, neglects the random nature of the heterogeneous mixture. Recently, MCNP has added a stochastic cell capability to its universe method (Brown, 2004). When a particle encounters a cell flagged as stochastic, the surfaces within that cell undergo a random translation, but are not placed beyond the boundaries of the enclosing cell. Each time a neutron enters a stochastic universe cell, a new random translation is made. This random "on-the-fly' assignment of geometry avoids the immense memory requirement of the benchmark explicit realization approach. This approach is illustrated in Figure 2.5 below for a 3×3 grid of coated fuel particle cells.



a) b)

Figure 2.5: Lattice Model of Repeated Geometry: a) Fixed Lattice and b) Stochastic Lattice

The stochastic lattice technique has the limitation that it restricts the randomness of problem geometry. In fixing the fuel particle coordinates to the lattice, there exists the possibility that if a neutron's path is aligned with a cell lattice boundary, it may stream indefinitely without interacting with a fuel particle. This will cause the distribution of matrix chord lengths to deviate from the distribution predicted by the Markovian mixing statistics assumption. The stochastic lattice method also allows for the possibility that the spatial distribution of material properties will change within a given particle history. A neutron that travels through a lattice cell, undergoes a scattering event and then later returns to that same lattice cell will find the fuel particle shifted to a different location. Thus, the stochastic lattice method is plagued by the same backscattering concerns as are present in the Levermore-Pomraning model and chord length sampling. However, MCNP stochastic lattice methods have thus far shown reasonable agreement with benchmark data and numerical comparisons (DiFilippo, 2003; Brown, 2004; Ji, 2004).

By restricting the spatial variance of the random geometry, the stochastic lattice method mitigates the impact of such geometry conflict errors.

2.5 SUMMARY

The main conclusion to be drawn from this survey of techniques is that explicit realization or benchmark calculations for performing Monte Carlo neutron transport simulations in the random media transport problems of interest in nuclear engineering are very computationally intensive. All approximate models designed to circumvent this computational expense including atomic mix, effective homogenization, chord length sampling, and stochastic lattice approaches, require essential assumptions in order to simplify the random media. Inevitably there are cases in which those assumptions break down.

The benchmark, atomic mix, chord length sampling methods are the computational techniques examined in further detail. The effective homogenization approach was excluded because its application is very system and geometry specific. In contrast, the MCNP lattice approach is applicable to a much wider array of transport problems. However, the fixed lattice approach has been thoroughly developed and tested for analysis of traditional nuclear reactor fuel assembly geometries. The newer MCNP stochastic lattice technique is intuitive and straightforward, but also well-developed within its own framework. The atomic mix and chord length sampling techniques should provide the most plentiful opportunities for potential method advancement. Both techniques are widely used, but each has known weaknesses. A more comprehensive understanding of the ranges of applicability of these two techniques warrants further investigation.

The initial focus of this work is to implement and compare numerical results from benchmark, atomic mix, and chord length sampling Monte Carlo algorithms. The benchmark explicit geometry realization simulations are performed to yield baseline results for comparison with the approximate techniques. The following section discusses considerations in developing a uniform computational framework for evaluating the merits of each of these methods.

Chapter 3: Code Development

This chapter describes the development of a Monte Carlo code package for simulating neutron transport in binary random heterogeneous mixtures. The purpose of this effort is to create a common framework for performing transport simulations in random heterogeneous materials using a variety of approximate computational models. In addition to establishing a common platform utilizing equivalent data and mathematical formulas, the development of an independent transport code enables much flexibility in accommodating future modifications including revised geometry models, material cross section data, and transport algorithms. Given the complexity of existing Monte Carlo transport packages, performing similar modifications to established code packages would be prohibitively difficult. The primary reference model under investigation in this work is an HTGR fuel pebble consisting of a graphite matrix and low enriched uranium dioxide fuel kernels (Johnson, 2001). The HTGR pebble is a sphere with radius 3.7959 cm. At the center of the pebble is a 2.5 cm fuel region containing 9,394 TRISO coated fuel particles randomly dispersed within a graphite matrix. The stochastic heterogeneity of the fuel pebble is modeled as 9,394 0.0251 cm radius uranium dioxide spherical fuel kernels randomly located within the graphite. For the studies performed within this research, the four thin coating layers surrounding the uranium dioxide kernel have been homogenized with the surrounding graphite matrix. Ji (2004) has shown that homogenizing the carbon based coating layers of TRISO particles with the rest of the graphite matrix yields negligible errors. Two outer regions, a graphite shell and representative moderator shell, surround the central fuel region. Figure 3.1 illustrates the

arrangement of the low enriched uranium pebble benchmark unit cell. The dimensions and material composition of this benchmark problem are given in Table 3.1 below.

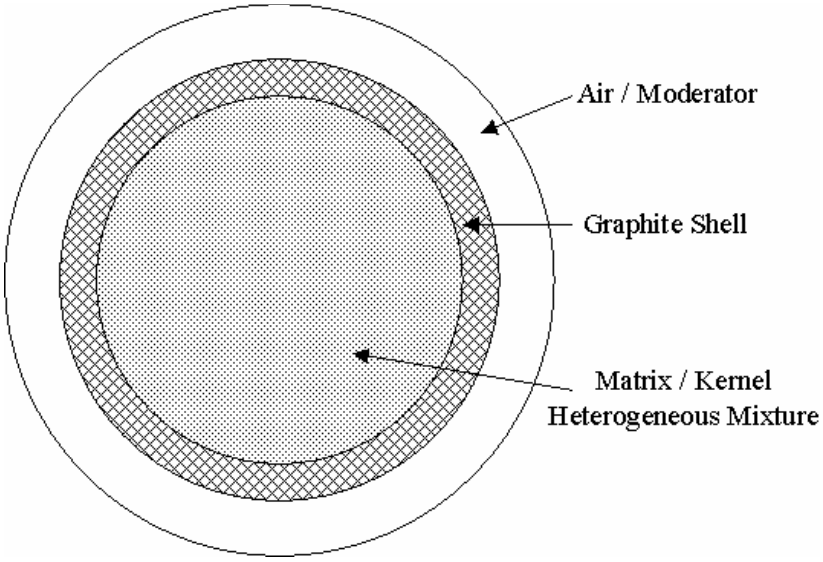


Figure 3.1: LEUPRO Pebble Fuel Element

Table 3.1 Material Specifications for LEUPRO Benchmark Problem (Johnson *et al.*, 2001)

Region	Volume (cm^3)	Nuclide	Atom Density (nuclei $barn^{-1}$ cm^{-1})
UO_2 kernel	6.62384E-05	^{16}O	4.86157E-02
		^{235}U	4.11729E-03
OD= 0.0502 cm		^{238}U	2.01906E-02
		Total	7.29236E-02
Buffer	1.02056E-4	^{10}B	1.58108E-08
		^{11}B	6.40418E-08
OD= 0.0685 cm		C (natural)	5.51511E-02
		Total	5.51511E-02
Inner PyC	6.59352E-05	^{10}B	2.73095E-08
		^{11}B	1.10617E-07
OD= 0.07648 cm		C (natural)	9.52609E-02
		Total	9.52610E-02
SiC	7.10387E-05	^{10}B	1.38E-08
		^{11}B	5.58080E-08
OD= 0.08354 cm		C (natural)	4.80603E-02
		Si (natural)	4.80603E-02
		Total	9.61207E-02
Outer PyC	9.63663E-05	^{10}B	2.73095E-08
		^{11}B	1.10617E-07
OD= 0.09154 cm		C (natural)	9.52609E-02
		Total	9.52610E-02
Matrix	6.56779E-03	^{10}B	2.47855E-08
		^{11}B	1.00394E-07
Side= 0.1910142 cm		C (natural)	8.64563E-02
		Total	8.64564E-02
Graphite Shell	47.64749	^{10}B	2.47855E-08
		^{11}B	1.00394E-07
ID= 5.0 cm		C (natural)	8.64563E-02
OD= 6.0 cm		Total	8.64564E-02
Moderator / Air	116.00678	^{1}H	2.97225E-07
Mixture		^{2}H	4.45905E-11
		^{10}B	1.17151E-08
OD= 7.5918 cm		^{11}B	4.74520E-08
		C (natural)	4.08644E-02
		^{14}N	2.04260E-05
		^{15}N	7.50350E-08
		^{16}O	5.12530E-06
		Total	4.08904E-02

The remainder of this chapter describes the key features and challenges of implementing a code package for performing Monte Carlo simulations in binary random heterogeneous mixtures. The code package developed in this work includes benchmark, atomic mix, and chord length sampling algorithms. A MATLAB routine has been coded to perform both fixed source and eigenvalue criticality calculations. MATLAB's built in random number generators are used for sampling the random variables that define stochastic process of the neutron transport simulation. The Monte Carlo tracking simulations performed in this work are strictly analog. There are no provisions for implicit capture, particle splitting or rouletting, or other variance reduction techniques.

3.1 GEOMETRY REPRESENTATION

Much of the power of the Monte Carlo method lies in its ability to accurately represent almost any complex arbitrary geometry. In almost every Monte Carlo calculation involving complex geometry, the neutron tracking through this geometry can be the most time consuming aspect of the entire simulation. For this work, the geometry considered consists of many non-intersecting spheres randomly located within a system of concentric bounding spheres. The geometry description, neutron locations, and direction vectors are defined on a rectangular Cartesian coordinate system.

The first step in the execution of a benchmark explicit tracking algorithm is to create an explicit realization of the fuel pebble geometry. This is done using a process known as random sequential addition (Widom, 1966). Randomly placed fuel kernel spheres are successively added to the pebble volume ensuring no overlap of previously placed kernels or of the pebble boundary.

Coated fuel particle sphere locations are sampled at random using a power law source distribution. All spherical coated fuel particles of radius r_0 must fit within a pebble of radius R. The locations of each of the sphere centers are most efficiently sampled in spherical coordinates in terms of r, the sphere's distance from the origin, the polar angle θ, and the azimuthal angle ϕ. The differential volume element occupied by the sphere center can then be defined as

$$dV = (r\sin\theta d\phi)(rd\theta)(dr) = r^2 dr\sin\theta d\theta d\phi \tag{3.1}$$

The probability distribution functions describing these coordinates are then

$$\begin{aligned} f(r) &= r^2 dr \\ f(\theta) &= \sin\theta d\theta \\ f(\phi) &= d\phi \end{aligned} \tag{3.2}$$

We desire to sample each of these coordinates using a random number ξ where $0 \le \xi \le 1$. For any continuous random variable x we can equate this random number to the cumulative distribution function $F(x)$ according to

$$F(x) = \int_a^b f(x)dx = \xi \tag{3.3}$$

where a and b are the minimum and maximum permissible values of x. In this case we find

$$\begin{aligned} r &= \sqrt[3]{\xi_1}(R - r_0) \\ \theta &= \cos^{-1}(1 - 2\xi_2) \quad \text{(Pevey, 2007).} \\ \phi &= 2\pi\xi_3 \end{aligned} \tag{3.3}$$

Only three random numbers are required to select the unbiased placement of each coated fuel particle. This manner of sampling is more efficient than standard rejection sampling techniques in modeling the random geometry.

To transform the selected location back to our standard Cartesian coordinate frame

$$x = r\sin\theta\cos\phi$$
$$y = r\sin\theta\sin\phi \tag{3.4}$$
$$z = r\cos\theta$$

For tracking a neutron's flight, position and direction are defined using standard (x, y, z) spatial coordinates and (u, v, w) direction cosine vectors. The straight-line flight of a neutron from point (x_1, y_1, z_1) to point (x, y, z) a distance s away can be described as

$$x = x_1 + us$$
$$y = y_1 + vs \tag{3.5}$$
$$z = z_1 + ws$$

In the standard rectangular coordinate system, the surface of a given sphere located at point (x_0, y_0, z_0) of radius R can be defined as all (x, y, z) triplets satisfying

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2 \tag{3.6}$$

Finding the point of intersection of this line segment and nearest sphere surface is the most frequent calculation performed in the benchmark simulation.

3.2 NUCLEAR DATA

Although the most often noted advantage of the Monte Carlo method is its flexibility in modeling arbitrary geometries, it is equally as versatile in its ability to accurately represent complicated material compositions. A complete description of the flight paths and interactions between the neutron and the nuclei of the medium is critical to the modeling of any physical transport process. By implementing a continuous energy description of material interaction probabilities, sharp variations in material cross sections can be represented with a high degree of fidelity. Thermal, resonance, and unresolved resonance effects can all be simulated using Monte Carlo techniques.

All cross section data was obtained from the ENDF (Chadwick *et al*., 2006) data libraries. Formatting of the data libraries was performed using the NJOY99 (MacFarlane, 1994) nuclear data processing code package. NJOY is a modular code package that prepares cross section data by first extracting the appropriate isotope information from the raw ENDF data libraries and then accounting for Doppler broadening, energy self-shielding, unresolved resonances, and other effects. NJOY reconstructs the cross section information into an output format that is more useful in a computational transport code than the raw ENDF data.

The energy group structure is broken into bins of equal lethargy. Neutron moderation is more appropriately modeled as linear in lethargy space than in traditional energy space. Lethargy, the logarithmic energy loss, u, is defined as

$$u = \ln \frac{E_{\max}}{E} \tag{3.7}$$

where $E_{\max}$ is the highest neutron energy within the system. Here $E_{\max}$ is taken to be 10 MeV. This is the uppermost energy considered within this work. The group structure uses 10 bins per decade for thermal energies below 1 eV and 100 bins per decade for energies between 1 eV and 10 MeV for total of 740 energy groups. This energy structure should be fine enough to approach the fidelity of a fully continuous energy data treatment without the expense of linking to continuous energy data. This finely resolved multi-group structure also decreases the dependence on the assumed intra-group weighting functions used in the NJOY cross section preparation. This is the same energy group structure as is used in the VBUDS nuclear fuel cycle simulation code (Schneider, 2002). All cross section data was processed a temperature of 300 K. This is the temperature at which all published benchmark results were carried out. Energy dependent elastic scattering, radiative capture, and fission cross sections were obtained for all isotopes listed in Table 3.1. Additionally, inelastic scattering was considered for uranium isotopes

^{235}U and ^{238}U. Inelastic scattering is a dominant energy loss mechanism for such heavy nuclei. For a particular target nucleus and neutron energy, the total cross section was taken to be the sum of all possible reactions. Other essential energy and nuclide dependent data included $\bar{\mu}$, the average scattering cosine and $\bar{\nu}$, the average number of neutrons emitted per fission. A sample NJOY input listing is given in Appendix A.

3.3 NEUTRON TRACKING

With the problem geometry and nuclear data established, the basic steps in simulating the neutron transport with the Monte Carlo method can be reduced to the following steps:

1. Select a source neutron
2. Determine the collision location
3. Determine the target nuclide and type of collision
4. Determine the result of collision
5. Tally results
6. Repeat until the history terminates by leakage or capture (Dupree and Fraley, 2002).

3.3.1 Source Definitions

Two types of source distributions were employed in this work. A single isotropically emitting point source was considered for leakage or transmission calculations. The spatial location of this source type is fixed for all histories. The initial direction cosines are randomly assigned according to

$$\begin{aligned} u &= \sin\theta\cos\psi \\ v &= \sin\theta\sin\psi \\ w &= \cos\theta \end{aligned} \tag{3.8}$$

where

$$\theta = 2\xi_1 - 1$$
$$\psi = 2\pi\xi_2 \tag{3.9}$$

The second type of initial source distribution is a uniform fission source that is used for eigenvalue criticality calculations. The assumption used for the initial fission neutron distribution, or first batch, is that the fission locations are uniformly distributed within the coated fuel particles. The first several neutron generations, or batches, are discarded to allow the spatial distribution of fission sites to settle from the initial uniform distribution to its steady-state, fundamental mode.

The second part of the eigenvalue source specification is the sampling of appropriate fission neutron energies. Fission neutron energies are selected using an energy dependent Watt fission spectrum. The Watt spectrum describes the fraction of prompt fission neutrons emitted at unit energy E, $\chi(E)$. The spectrum can be well approximated as

$$\chi(E) = Ae^{-e/B} \sinh\sqrt{CE} \tag{3.10}$$

where A, B, and C are fitted constants. This spectrum is sampled using the rejection technique developed by Everett and Cashwell (Dupree and Fraley, 2002).

3.3.2 Free-Flight Determination

Neutrons travel in a straight line between collisions. The free-flight distance is sampled using the macroscopic total cross section Σ_t where

$$\Sigma_t = \sum_i N_i \sigma_{t,i} \tag{3.11}$$

where $\sigma_{t,i}$ is the total microscopic cross section of each material at the neutron's current energy.

Within a homogeneous region, the probability of interaction is constant and proportional to the total macroscopic cross section. The probability that a neutron will have a collision while moving a distance of ds is

$$dP = \sum_t ds \tag{3.12}$$

The probability of a neutron travels a distance s without interaction is

$$P(s) = e^{-\Sigma_t s} \tag{3.13}$$

The probability that a neutron travels a distance s and then has a collision within ds is

$$P(s+ds) = \sum_t e^{-\Sigma_t s} \tag{3.14}$$

This is the PDF for the collision distance. The cumulative distribution function (CDF) is then

$$P(s) = \int_0^s \sum_t e^{-\Sigma_t s'} ds' = 1 - e^{-\Sigma_t s} = \xi \tag{3.15}$$

It then follows

$$s = \frac{-1}{\sum_t} \ln(1-\xi) = \frac{-1}{\sum_t} \ln(\xi) \tag{3.16}$$

Sampling a random value of s from this distribution is among the most frequent of all operations performed in the Monte Carlo simulation.

3.3.3 Collision Sampling

The target nuclide and reaction type can both be represented as sets of discrete probabilities. The total macroscopic cross section is the sum of the cross sections of all isotopes present within the material. The probability that the neutron has a collision with a particular target nuclide j is

$$p_j = \frac{N_j \sigma_{t,j}}{\sum_i N_i \sigma_{t,i}} \tag{3.17}$$

Then to select the target nuclide we sum the collision probabilities such that

$$P_j = \sum_{i=1}^{j} p_i \tag{3.18}$$

Discrete sampling can be used such that $P_{j-1} \le \xi \le P_j$ (Brown, 2005).

Similarly, to select the reaction type, we consider the discrete probability of each possible reaction for that particular isotope.

$$\sigma_t = \sigma_{elastic} + \sigma_{inelastic} + \sigma_{capture} + \sigma_{fission} \tag{3.19}$$

The probability of a particular reaction j is then

$$p_j = \frac{\sigma_j}{\sigma_t} \tag{3.20}$$

Discrete sampling is used again to sample the discrete reaction type probabilities. An accurate stochastic description for each of the possible neutron collision mechanisms is essential for a successful Monte Carlo simulation. The following paragraphs discuss how the outcome of each of the possible reaction types listed in equation is determined using random sampling.

Scattering collisions result in a change in the energy and direction of the incident particle. Elastic scattering is one of the dominant forms of energy loss for fast neutrons slowing down to low energies. In this type of reaction the total kinetic energy of the neutron-nucleus collision is conserved. Elastic scattering is most easily handled using two frames of reference. The lab frame is the actual coordinate frame used for the neutron tracking and is attached to the fuel element geometry. A second frame of reference, the center of mass frame is affixed at the point of the collision and is used purely to describe the kinematics of the collision. Figure 3.2 illustrates an elastic collision in the lab frame.

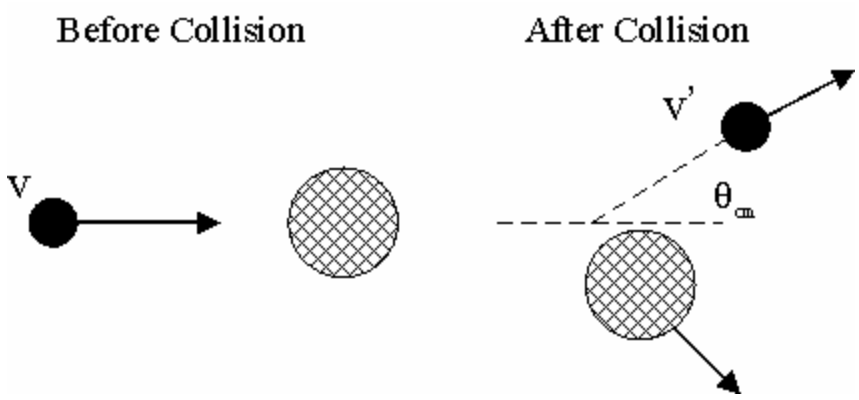


Figure 3.2: Center of Mass Scattering Kinematics for Incoming Neutron Velocity v and a Stationary Target Nucleus

For most neutron energies the target nucleus can be well approximated as being at rest in relation to the colliding neutron. The scattering can then be completely defined by the center of mass scattering angle θ_{cm}. Elastic scattering is assumed isotropic in the center of mass frame. To simulate the elastic collision, a center of mass scattering angle is sampled using equation 3.9 above. The outgoing neutron energy, E' can then be calculated from conservation of energy and momentum considerations as

$$E' = E\frac{A^2 + 2A\cos\theta_{cm} + 1}{(A+1)^2} \tag{3.21}$$

where A is the atomic mass of the target nucleus. One important note from this expression is that the neutron cannot gain energy as the result of a collision with a stationary nucleus.

Figure 3.3 shows the relationship between the neutron scattering angle in lab frame and center of mass frame coordinates.

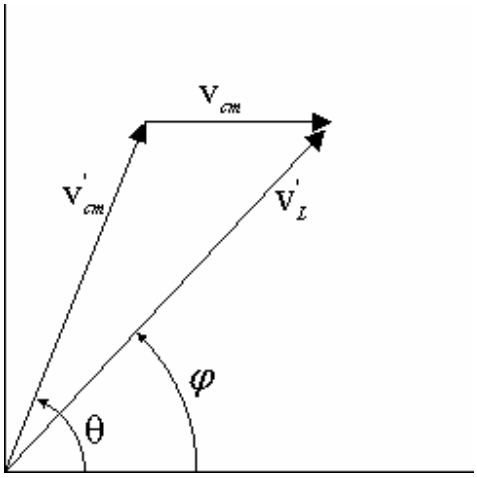


Figure 3.3: Rotation from Center of Mass Frame to Lab Frame Scattering Angle (Dupree and Fraley, 2002).

From this figure we see

$$v_L^{'}\cos\varphi = v_c^{'}\cos\theta + v_{cm} \tag{3.22}$$

where $v_L^{'}$ is the outgoing neutron velocity in the lab frame, $v_c^{'}$ is the outgoing particle velocity in the center of mass frame, and v_{cm} is the velocity of the center of mass. The lab frame scattering angle, φ, can be expressed as a function of center of mass scattering angle, θ as

$$\cos\varphi = \frac{1 + A\cos\theta}{\sqrt{A^2 + 2A\cos\theta + 1}} \tag{3.23}$$

The final step in the elastic collision simulation process is to rotate from the incoming direction vector (u, v, w) to the outgoing direction vector (u', v', w'). Setting $\mu_{lab} = \cos\varphi$ and sampling a polar scattering angle $\phi = 2\pi\xi$, the outgoing direction cosines (u', v', w') may be defined as (Brown, 2005)

$$
\begin{aligned}
u' &= u\mu + \frac{\sqrt{1-u^2}\left(uw\cos\phi - v\sin\phi\right)}{\sqrt{1-w^2}} \\
v' &= v\mu + \frac{\sqrt{1-u^2}\left(vw\cos\phi + u\sin\phi\right)}{\sqrt{1-w^2}} \\
w' &= w\mu - \sqrt{1-u^2}\sqrt{1-w^2}\cos\phi
\end{aligned}
\tag{3.24}
$$

Neutron scattering at thermal energies is more complex. Target nuclei are always in motion due to thermal energy. However, for the bulk of the of the neutron slowing down process, the speed of the neutrons is much greater than that of the atoms within the moderating medium and thus thermal motion of target nuclei can be ignored. When the speed of the neutron is comparable to the speed of the target nuclei, the target can no longer be well approximated as being stationary. The "effective" elastic scattering cross section and the neutron's outgoing direction and energy are all influenced by the thermal motion of the target nuclei. At thermal energies, the neutron is equally as likely to gain energy as is it to lose energy in an elastic scattering collision. Thus, for energies below approximately 1 eV, the scattering simulation requires an additional sampling of the target nucleus velocity.

The collision kinematics are sampled in the zero temperature, stationary target center of mass frame (Brown, 2003). The outgoing neutron energy and direction are adjusted to account for the relative velocity of the incident colliding particles. This work uses the free gas model target nuclei velocity and rejection sampling technique described by Dupree and Fraley (2002). For graphite, chemical binding effects of the graphite lattice are not well treated by the free-gas model. A separated graphite scattering kernel is utilized for all graphite collisions at energies below 1 eV.

Inelastic scattering is a significant energy loss mechanism for high energy neutrons impinging on heavy targets. In this type of reaction, the neutron is absorbed by

the target nucleus to form a compound nucleus. The compound nucleus then decays by emitting a neutron, but may be left in an excited state. Unlike elastic scattering, there are no conservation of energy or momentum relationships that describe this process. Thus all outcomes of inelastic scattering reactions are sampled directly from discrete inelastic scattering kernel probability tables.

In the event of a capture reaction, the neutron history is terminated at that interaction point. The code developed here provides strictly for neutron transport studies, thus secondary photon production is not considered.

The second type of possible absorption reaction is fission. Fission, like capture, results in the termination of the neutron history. Each fission event results in the emission of some number of fission neutrons, $\bar{\nu}$. $\bar{\nu}$ is a tabulated value that depends on the target nucleus and the incident neutron energy. For eigenvalue criticality calculations, the location of the fission and number of neutrons produced is recorded for assessing the criticality of the system.

3.4 TALLIES AND ESTIMATORS

A Monte Carlo calculation generates results by tallying events of interest such as surface crossings, collisions, or fissions. The contribution that a particular history makes to this tally is computed using an estimator.

The most frequently used estimator within this work is known as the collision estimator. The collision rate is

$$R = \sum_t \phi \tag{3.25}$$

The flux contribution within a particular region can therefore be determined by summing $1/\sum_t$ for all collisions and over all histories.

$$\phi = \frac{1}{\tilde{V}N} \sum_{\substack{\text{all} \\ \text{histories}}} \frac{1}{\Sigma_t} \tag{3.26}$$

Here, Σ_t is the macroscopic total cross section at the point of collision, N is the total number of neutron histories in the simulation, and $\tilde{V}$ is the volume of the region. This estimator can be applied to any reaction rate of interest. Similarly, a path length flux estimator is also used for the estimation of the average scalar flux across a particular region of interest. Each time a neutron history travels across any part of a tally region, the segment length is recorded.

$$\phi = \frac{1}{\tilde{V}N} \sum_n l_n \tag{3.27}$$

where l_n is the path length of each particle segment tracked across the region of interest. The path length estimator is inherently more efficient at determining the flux within very small geometry regions (Leppanen, 2007). This is because the collision estimator requires the particle to actually collide within the volume. If a particle traverses the region and undergoes no collisions, it does not make a tally contribution according to the collision estimator. In contrast, the path length estimator does not require the neutrons to undergo an interaction in order to make a tally contribution.

In criticality calculations, the multiplication factor of the system is the variable of primary interest. The multiplication factor is the number of neutrons at the end of a generation divided by the number of neutrons beginning the generation. Because such calculations consider the neutron progeny, k, the number of neutrons produced per fission reaction must be taken into account. Three types of fission event estimators were employed within this work. The product $\bar{\nu}\Sigma_f$ gives the number of fission neutrons per path length of neutron flight. A path length estimator for the multiplication is then

$$k_{pathlength} = \frac{1}{N} \sum_n l_n \bar{\nu} \Sigma_f \tag{3.28}$$

Similarly, collision and absorptions estimators can be used by considering only collision or absorption events with fissile isotopes.

$$k_{collision} = \frac{1}{N}\sum_{n}\frac{\bar{\nu}\Sigma_f}{\Sigma_t} \tag{3.29}$$

$$k_{absorption} = \frac{1}{N}\sum_{n}\frac{\bar{\nu}\Sigma_f}{\Sigma_a} \tag{3.30}$$

Criticality calculations were performed using the ratio of generations method. A group, or batch, of neutrons histories is simulated, the multiplication factor, k, is recorded and the fission neutron progeny is re-normalized and randomly sampled to select the source of fission neutrons for the next generation. The final estimate for k is attained by averaging the individual generation k estimates over many successive neutron generations.

Infinite medium, or k_∞, criticality calculations of HTGR fuel pebble elements are performed assuming a white boundary condition. This means that neutron histories that encounter the fuel pebble surface are reflected with a uniform cosine directional distribution. This boundary condition approximates a uniform isotropic flux field. Leakage from the unit fuel pebble is not permitted and the history can only be terminated by the occurrence of an absorption reaction.

3.5 BENCHMARK ALGORITHM

A benchmark solution is achieved by averaging transport solutions over many different physical realizations each having the nominal mixing statistics. Each physical realization is generated using random sequential addition. The routine of tracking a particle within the problem geometry is essentially a repetition of the following three calculations:

i) calculating distance to the pebble boundary, where leakage or reflection occurs,

ii) calculating the distance to the closest material interface,

iii) and sampling the distance to a collision

The shortest of these three distances is used to designate the location of the next event and the particle's new location is updated. If the neutron encounters a boundary, either the neutron leaks from the system or is reflected. If a material interface is encountered, the appropriate material cross sections are updated and the minimum distance routine is repeated within the new material. If a collision event occurs, the target nuclide and reaction type is sampled and the outcome of the collision is determined. A collision that results in fission or capture event will cause the neutron history to be terminated.

Because each physical realization represents only one of an infinite possible number of combinations of the random heterogeneous media, the entire calculation is repeated a number of times to obtain an overall, or ensemble averaged result. This measure reduces the potential errors that may be attributed to the random heterogeneity effects of a single particular geometrical realization.

$$\overline{X} = \frac{1}{N_{realizations}} \sum_{r=1}^{N_{realizations}} \overline{x}_r \tag{3.31}$$

The vast majority of the computation expense of the benchmark analysis of the LEUPRO pebble results from calculation ii). Because of the random heterogeneity, for a neutron residing within graphite matrix, the distance to closest interface calculation must consider all 9,394 coated fuel particles. This calculation must be repeated following every matrix collision. Initial k_∞ calculations of the LEUPRO pebble showed more than 98% of the total simulation time is spent determining the distance to the closest coated fuel particle.

3.5.1 Delta Tracking

A technique that has proven to accelerate the benchmark analyses within this work is delta tracking, otherwise known as Woodcock tracking or hole tracking. Typically the free flight distance is sampled according to the material total cross section.

$$s = \frac{-\ln(\xi)}{\Sigma_t(E,\vec{r})} \tag{3.32}$$

Delta tracking instead uses an energy dependent maximum cross section Σ*.

$$\begin{aligned} \Sigma^*(E) &= \max\left(\Sigma_{t,1}(E,), \Sigma_{t,2}(E)\right) \\ \Sigma^*(E) &= \Sigma_1(E) + \Sigma_\delta^1(E) = \Sigma_2(E) + \Sigma_\delta^2(E) \end{aligned} \tag{3.33}$$

where $\Sigma_\delta^1(E)$ and $\Sigma_\delta^2(E)$ are additional artificial, or delta, cross sections. The distance to the next collision using delta tracking is sampled using this maximum cross section.

$$s' = \frac{-\ln(\xi)}{\Sigma^*(E)} \tag{3.34}$$

At the collision point we must determine whether the collision is a real collision or is due to the delta component of the maximum cross section. At the collision point, the material total cross section $\Sigma_t(E)$ is identified. There is a finite probability $\Sigma_t(E)/\Sigma^*(E)$ that the collision has occurred. Otherwise, the sampled collision is due to the artificial cross section, $\Sigma_i^\delta(E)$. If $\xi \le \Sigma_t(E)/\Sigma^*(E)$ the collision is real. If a pseudo, or delta collision is sampled the neutron undergoes no change in energy or direction. In this instance, the free-flight distance from the delta collision point is re-sampled using the maximum cross section until a real collision occurs. The additional delta collisions do not alter the distribution of real collisions (Smith, 1994).

Effectively, the geometry boundaries that separate the spatial regions are ignored until a collision occurs. Thus, delta tracking can be very useful in complex geometries in which the material cross section varies rapidly over the neutron flight path. Delta tracking capabilities have been implemented in a wide array of Monte Carlo codes

(Smith, 1994; Sutton, 1999). There are two major drawbacks to delta tracking. Path length tallies can not be used because the points at which a neutron crosses geometry interfaces are not recorded. The length of track length segments across a particular region within the heterogeneous mixture is not available. Therefore, only collision estimators may be used. Secondly, delta tracking can be highly inefficient if the macroscopic total cross sections of the constituent materials are very dissimilar. Such an instance is illustrated in Figure 3.4. In this case, there will be a great deal of computational overhead devoted to examining a large number of delta collisions between each real collision.

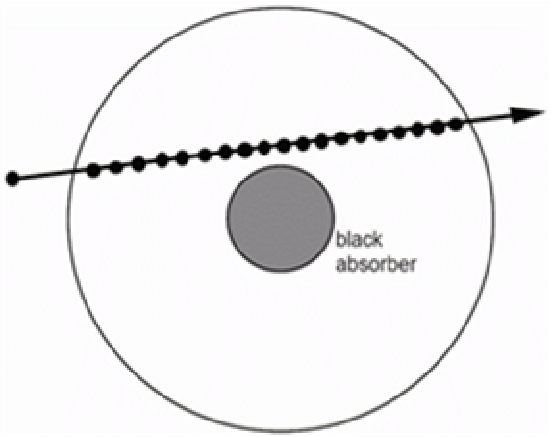


Figure 3.4: Inefficient Application of Delta Tracking in which many Delta Collisions are Sampled between each Real Collision (MONK, 2003)

This difficulty arises most frequently in problems involving highly localized absorbers and at neutron energies in the thermal or resonance range. In these energy ranges the nuclear cross section exhibits the greatest variance between isotopes. Some Monte Carlo codes include special provision for counteracting this inefficiency. RACER (Sutton, 1999) allows for the specification of either standard or tracking to be performed

within particular energy ranges. Similarly Leppanen (2007) implemented a tracking algorithm in which the macroscopic total cross sections of particular localized heavy absorbers could be removed from delta tracking considerations. This precludes the inefficiencies noted above, but also introduces an additional distance-to-heavy absorber calculation between each collision segment.

Standard delta tracking was applied to the heterogeneous coated fuel particle and graphite mixture within the LEUPRO pebble k_∞ benchmark. Despite the inefficiencies noted above, delta tracking yielded an overall speedup of the computation time per history of approximately 54. The decrease in computation time occurs because the immense number of material boundaries are only considered in the event of a collision. Although inefficiencies exist in parts of the tracking algorithm as described above, there is a substantial net improvement in computational efficiency for the LEUPRO benchmark k_∞ calculation.

3.6 ATOMIC MIX ALGORITHM

The atomic mix routine does not require multiple realizations of the pebble geometry because the fuel and matrix are assumed to coexist as a single homogeneous mixture. All cross sections are assumed to be volume weighted averages of all the constituent materials within the region. Any macroscopic cross section $\sum_i$ has an expected value of

$$\left\langle \sum_i \right\rangle = V_0 \sum_{i,0} + V_1 \sum_{i,1} \tag{3.35}$$

The two outermost shells are not included in the homogenization. In reducing the material description of the fuel-matrix mixture to a single set of homogenized cross sections, the atomic mix routine eliminates the need to rigorously calculate the distance to

the nearest material interface within the heterogeneous benchmark model. The atomic mix neutron tracking algorithm condenses to just two basic calculations: calculating distance to the pebble boundary, where transmission or reflection occurs, and sampling the distance to next collision. Collision physics and outcomes are sampled using the homogenized cross sections.

3.7 CHORD LENGTH SAMPLING ALGORITHM

Chord length sampling is performed using the limited chord length sampling (LCLS) method of Donovan and Danon (2003b). Using this method, chord length sampling within the matrix material replaces the benchmark method of explicitly calculating the location of the closest material interface.

For a uniform, random distribution of equal sized spheres, the probability distribution of matrix chords is Markovian. This means that chords lengths between kernels have an exponential distribution. The probability of finding of a chord of length λ within the matrix is given by

$$p(\lambda) = \tfrac{1}{\bar{\lambda}} e^{-\frac{\lambda}{\bar{\lambda}}}. \tag{3.36}$$

where $\bar{\lambda}$ is the average chord length in the matrix. Thus, the average chord length must first be known in order to sample from the distribution given in equation 3.36. For an infinite, random arrangement of three-dimensional spheres of radius r, the average matrix chord length is

$$\bar{\lambda} = \frac{4 \cdot VF_{matrix}}{3 \cdot VF_{spheres}} r \quad \text{(Torquato, 2002).} \tag{3.37}$$

However, this formula becomes non-exact when one considers the finite dimension of the pebble. The average chord length must be modified to account for the finite dimension of the stochastic region. All coated fuel particles spheres are constrained

to fit entirely within a bounding spherical volume. This work uses Donovan's centerline non-overlap formulation for determining an empirical matrix average chord length in finite geometries (2003c). In this method, the effect of the finite dimension of the fuel pebble is correlated to the mean matrix chord length by generating a large number of random realizations and observing the probability that a neutron will encounter no interstitial spheres along its initial direction of flight before reaching the edge of the pebble.

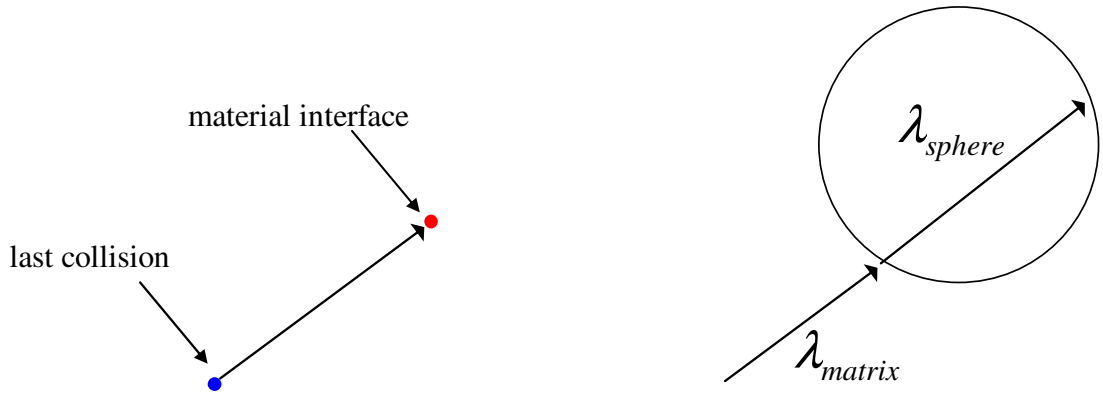
Once the fuel kernel sphere interface is located by sampling from the matrix chord length distribution, another chord length is sampled from a second distribution to assign the coordinates of the coated fuel particle. This second distribution is that of chord lengths within a convex sphere. The average chord length for a neutron traversing a sphere or radius r is

$$\bar{\lambda}_{sphere} = 4\frac{\text{volume}}{\text{surface area}} = \frac{4}{3}r. \tag{3.38}$$

A chord is randomly chosen according to

$$\lambda_{sphere} = \sqrt{4r^2\xi} \quad \text{(Donovan, 2003c).} \tag{3.39}$$

The assignment of kernel geometry is a two step procedure as illustrated in Figure 3.5. Once the sphere's coordinates are fixed, the explicit particle tracking algorithm is used until the particles exits the fuel particle.



Step 1. Sample matrix chord length to determine point where surface of sphere is encountered

Step 2. Sample sphere chord length to fix sphere coordinates

Figure 3.5: Assignment of Fuel Kernel Sphere Coordinates

Criticality calculations require an initial explicit physical realization of the fissioning coated fuel particle to be modeled. This ensures that all fission neutrons are born from within fissile coated fuel particle spheres. Similarly, successive generations require a banking of both fission sites and the corresponding sampled coated fuel particle coordinates.

3.8 PRELIMINARY RESULTS AND DISCUSSION

Each of the above three algorithms was implemented in MATLAB. Preliminary results validating the Monte Carlo procedure used in this work are given in Figures 3.6 and 3.7 below. Figure 3.6 presents k_∞ convergence results for the benchmark LEUPRO pebble as specified by Johnson (2001) for benchmark, atomic mix, and chord length sampling algorithms. Published benchmark results are approximately 1.722 for the fully

detailed explicit realization and 1.63 for the homogeneous model. Our results lie within 1% of these values.

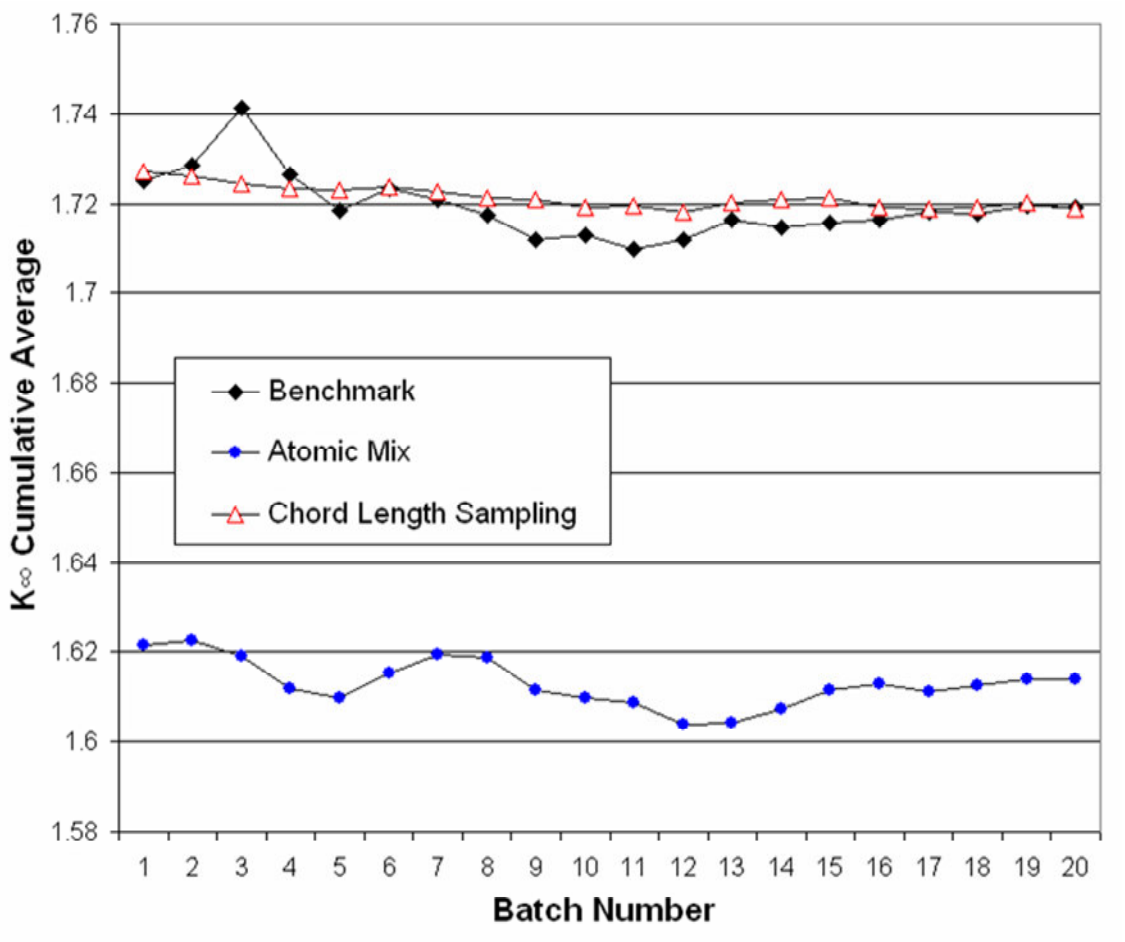


Figure 3.6: Comparison of k_∞ Results for LEUPRO Fuel Pebble

A comparison of the total computation time to track particles in a k_∞ calculation of a 9,394 kernel fuel pebble is summarized in Table 3.2. The data below represent the time to track a fission neutron from birth until capture averaged over many histories. The benchmark calculation is more than 250 times more time consuming than the homogeneous model on a per history basis. Within the benchmark algorithm using

standard tracking, more than 98% of the calculation time is spent determining the distance to the closest fuel kernel sphere. This very expensive distance calculation is eliminated with both the atomic mix and chord length sampling methods.

Table 3.2. Relative Time to Track Each History in the 9,394 Kernel LEUPRO Fuel Pebble k_∞ Calculation

Atomic Mix Homogenization	1
Chord Length Sampling	1.09
Benchmark (Standard Tracking)	256

Figure 3.7 presents a comparison of the neutron flux spectra within the LEUPRO fuel pebble benchmark problem as computed by MCNP and the MATLAB code developed in this research. The neutron flux spectra exhibit good agreement across the entire energy range. The only notable differences in the two spectra lie just above the thermal peak in the region of transition between zero-temperature purely elastic and thermal scattering. The physics models used here are somewhat less robust than those used in production level transport codes. Our thermal scattering treatment does not consider neutron energies above 1 eV, whereas MCNP takes into account thermal effects up to a few eV. At slightly higher energies, our code slightly under predicts the flux escaping the resonance range. This is most likely due to the assumed energy self-shielding treatment used in NJOY for the preparation of the multigroup data libraries.

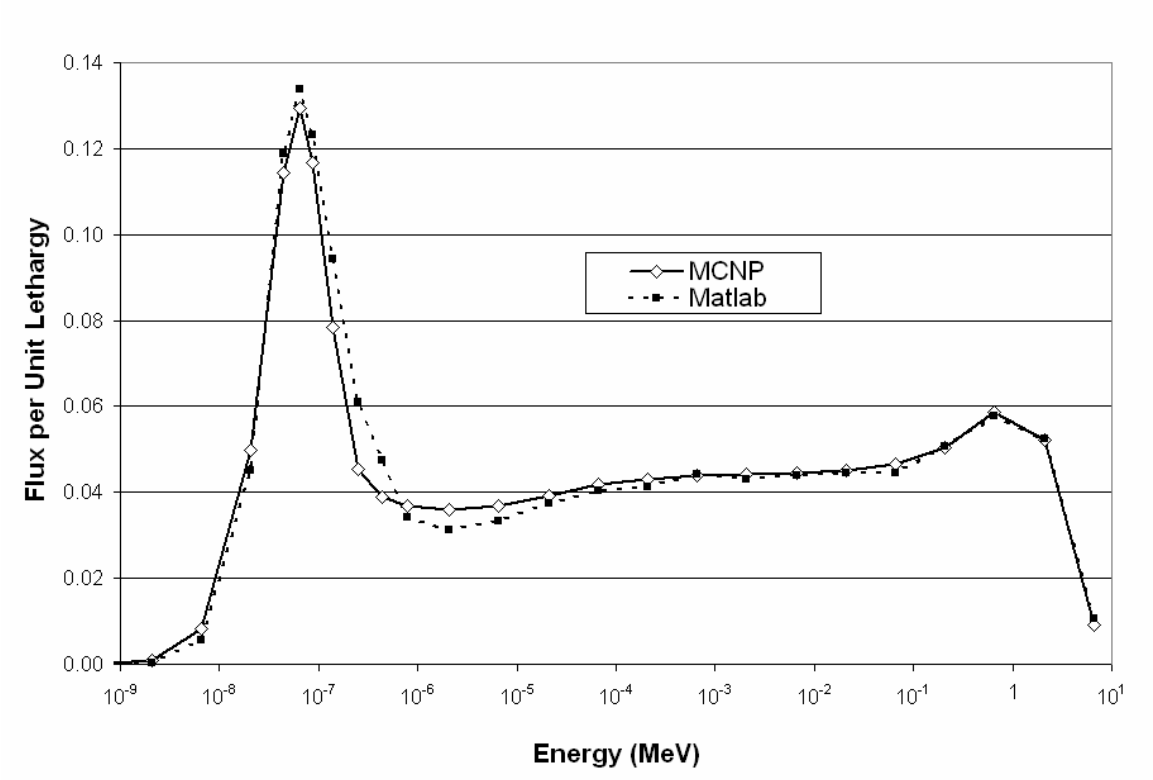


Figure 3.7: Comparison of Neutron Flux Spectra for LEUPRO Benchmark Fuel Pebble Atomic Mix Model

The previous illustrations demonstrate that the MATLAB implementation of the benchmark, atomic mix, and chord length sampling algorithms as described above do exhibit reasonable agreement with published benchmark data and results obtained with commercial nuclear analysis software. Commercial software packages may indeed perform these simulations in a more efficient or optimized manner. The goal of this research, however, was not to compete with these more established computer codes for the purpose of solving a specific representative or benchmark problem. With the flexible, easily adaptable computational package developed here, we are able to perform method

comparisons and also have the capability to incorporate future algorithm revisions as warranted.

Chapter 4: Error Parameterization

As exhibited by the initial studies, performing accurate transport calculations in random heterogeneous mixtures of interest in nuclear engineering applications can be a mathematically complex and computationally daunting task. A computational framework has been developed that will enable the examination of the relative costs and benefits of the benchmark, atomic mix, and chord length sampling Monte Carlo solution techniques. The suitability of each of these computational methods referenced is dependent upon the particular material and geometry configuration of the problem being analyzed. This chapter discusses the characterization of binary random heterogeneous media neutron transport problems in terms of critical dimensionless parameters. There are a number of factors that will affect the validity of a particular approximate method including geometry considerations, boundary effects, and energy dependence of cross sections. The ultimate goal of this effort will be to have a more comprehensive understanding of how such factors affect the suitability of approximate methods such as atomic mix and chord length sampling to model a specific random heterogeneous mixture transport problem.

Typically, the measure of success of a new or revised transport model is a comparison between the new model and benchmark calculation results for a few specific cases. Adams *et al*. (1989) used this approach as they examined an assortment of one-dimensional geometry cases in assessing the Levermore-Pomraning approximation. Donovan (2003c) performed a similar study with Monte Carlo techniques. This study focused on the applicability of chord length sampling in scattering conditions. Chord length sampling exhibited good agreement with benchmark results until the matrix optical scattering thickness parameter $\left(\sum_s \cdot \overline{\lambda}_{matrix}\right)$ exceeds 2. More recently Davis *et al*.

(2004) compared atomic mix and Levermore-Pomraning approximation results with benchmark results for a series of deterministic one-dimensional planar geometry calculations. This study showed the inadequacy of the atomic mix model when the dimensionless optical thicknesses of any of the material regions is large relative to a single mean free path. Davis also illustrated the inaccuracies of the Levermore-Pomraning approximation under scattering conditions. This work is the most comprehensive method comparison to date, but the study was restricted to one of one-dimensional analysis and limited the mixture composition by specifying that one of the constituent materials within the mixture is a void.

The primary comparison in this work is between the benchmark and chord length sampling techniques. The analysis presented herein attempts to establish a more generalized assessment of how all the material and geometry descriptive parameters affect the suitability of a solution method.

4.1 Description of Model for Parameter Studies

The model examined in this parameterization study is a simplified model of the LEUPOR geometry. It consists of a binary mixture within a sphere of radius 2 cm containing 400 randomly dispersed low enriched UO_2 spherical kernels of radius 1 mm. Borated graphite is the background matrix material. A vacuum boundary condition surrounds the pebble. All assumed atom densities are taken from the benchmark specifications given in Table 3.1. This model does not include the carbide coating layers surrounding the fuel kernels. The volume fraction of the mixture occupied by the UO_2 kernels is 5%. This packing density is comparable to the nominal volume fraction of coated fuel particles within the LEUPRO benchmark mixture of approximately 5.7%. The smaller number of UO_2 kernels allows for benchmark results to be generated in more

computationally feasible manner as compared to the 9,394 kernel LEUPRO benchmark specifications.

The source term considered is an isotropic point source located directly at the pebble center. Neutrons that leak from the outer boundary are tallied. Each neutron makes an equal contribution. The difference in the leakage fraction predicted by the benchmark algorithm and that predicted by chord length sampling is the primary metric for evaluation. Ensemble average leakage results are averaged over 1000 realizations each containing 1000 history simulations. It was observed that if the physical realization contains a UO_2 kernel residing at the source point, the leakage probability would be significantly affected. To account for this potential modeling discrepancy, both benchmark and chord length sampling simulations first begin with an explicit model of the random geometry to determine the material located at the source point.

One of the greatest challenges is in handling the cross section energy dependence. Material interaction probabilities for most isotopes of interest within nuclear engineering vary greatly with neutron energy. For this work, we collapse the multi-group cross section data to attain a physically representative one –group cross section set for our base case model. For each isotope *i* and reaction type *j* the one-group cross section $\sum_{j,1\,\text{group}}^{i}$ is defined according to

$$\sum_{j,1\,\text{group}}^{i} = \frac{\int_0^\infty dE \sum_j^i (E)\phi(E)}{\int_0^\infty dE \phi(E)} \tag{4.1}$$

$\phi(E)$ is the energy dependent scalar, or total, flux spectrum. This spectrum is obtained from a k_∞ simulation of this same reference problem. The pertinent macroscopic cross sections resulting from this group-collapse exercise are listed below in Table 4.1.

Table 4.1 Spectrum Average Macroscopic Material Cross Sections used in Parameterization Study

Interaction Type	Macroscopic Cross Section $\left(cm^{-1}\right)$
$\Sigma_{t,\ker nel}$	0.6117
$\Sigma_{s,\ker nel}$	0.5162
$\Sigma_{t,matrix}$	0.3464
$\Sigma_{s,matrix}$	0.3464

For this study the matrix material is assumed to be purely scattering. Thus, the only extinction mechanism is absorption within the interstitial UO_2 kernels. No distinction is made between fission and capture events. Subsequent calculations are conducted by linearly scaling the above nominal one-group cross section values.

Davis *et al*. (2004) identified seven key parameters in describing particle transport within binary heterogeneous media: a macroscopic scattering cross section, total macroscopic cross section, and mean chord length for each of the two materials, and a single overall characteristic length. In this work we consider a fixed characteristic length, namely a binary mixture pebble of radius 2 cm. We then examine the relative error of the chord length sampling approximation as a function of three dimensionless parameters: interstitial optical thickness $\left(\overline{\lambda}_{particle} \cdot \Sigma_{t,particle}\right)$, matrix scattering optical thickness $\left(\overline{\lambda}_{matrix} \cdot \Sigma_{s,matrix}\right)$, and mean chord length ratio (CLR). The interstitial optical thickness denotes the width of the interstitial kernel in units of neutron mean free paths. Similarly, the matrix optical scattering thickness is related to the probability of a neutron's free flight path between successive UO_2 kernels being interrupted by a scattering collision

within the matrix. The mean chord length ratio is a dimensionless description of the packing density of interstitial kernels.

$$CLR = \frac{\overline{\lambda}_{matrix}}{\overline{\lambda}_{kernel}} \tag{4.2}$$

For an infinite system of randomly arranged spheres in which an analytical expression for $\overline{\lambda}$ can be written for both matrix and interstitial, the chord length ratio reduces to

$$CLR = \frac{\overline{\lambda}_{matrix}}{\overline{\lambda}_{kernel}} = \frac{\frac{4}{3}\frac{VF_{matrix}}{VF_{kernel}}r}{\frac{4}{3}r} = \frac{VF_{matrix}}{VF_{kernel}} \tag{4.3}$$

Using the empirically determined value for $\overline{\lambda}_{matrix}$ the reference model has a CLR of 15.2. This compares to the CLR for the LEUPRO benchmark mixture of 15.0 (Donovan, 2003c).

4.2 RESULTS AND DISCUSSION

Figure 4.1 illustrates the accuracy of the chord length sampling method in predicting the leakage fraction from the reference pebble geometry under a variety of material configurations. The matrix scattering optical thickness parameter varied between 0.01 (very low scattering) and 10 (very high scattering). Interstitial cross sections were scaled to represent total optical thicknesses of 10 and 50 times the nominal spectrum average values. The interstitial kernel scattering ratio $\Sigma_{s,kernel}/\Sigma_t$ is constant for all cases. The relative errors are plotted as a function of both matrix scattering optical thickness and interstitial total optical thickness.

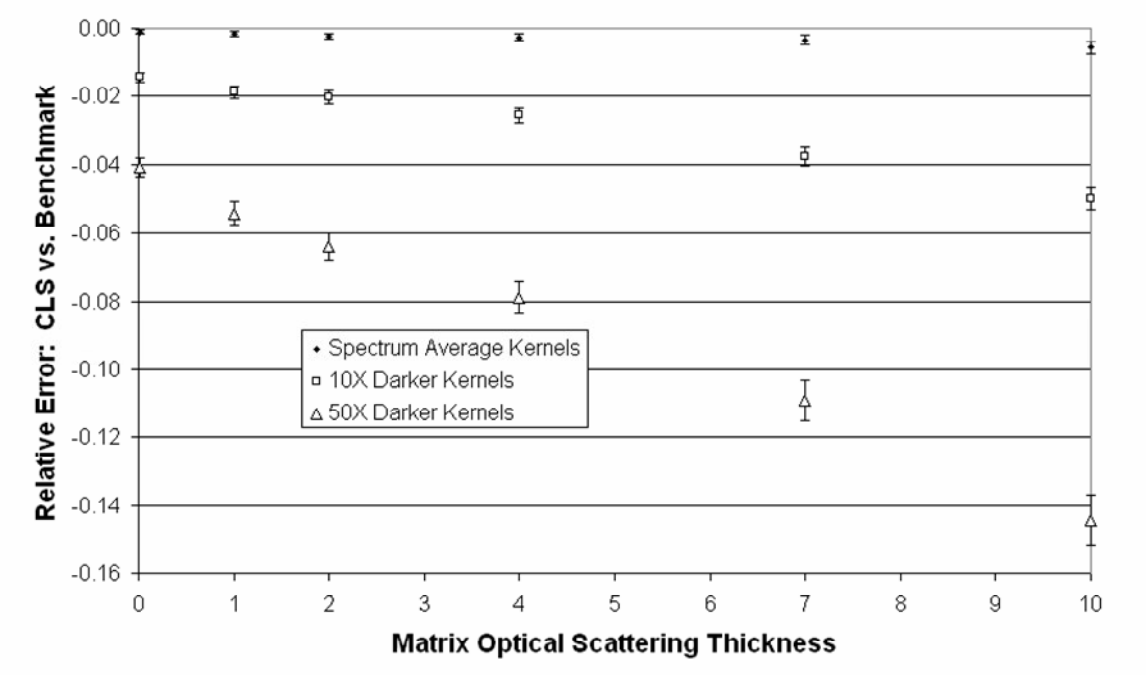


Figure 4.1: Relative Error of Chord Length Sampling Leakage Probability as a Function of Increasing Matrix Scattering and Interstitial Kernel Absorption

In all cases the chord length sampling method over-predicts the absorptive effectiveness of the mixture. The uncorrelated segments assumption of chord length sampling means that a neutron traversing through a heterogeneous mixture has no memory of its history. Therefore, a neutron may pass through a particular region, have a scattering collision, and travel backwards only to encounter a different material than it saw previously. This geometry modeling error is illustrated in Figure 4.2 below.

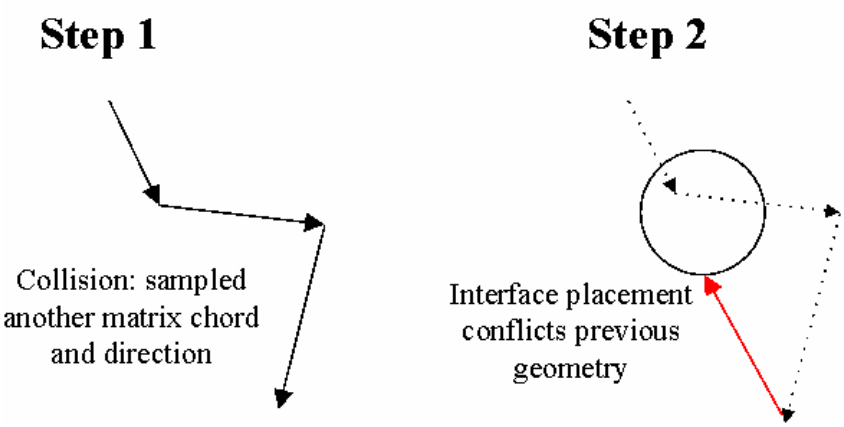


Figure 4.2: Placement of Chord Length Sampling Interfaces at Conflicting Locations

The net effect of chord length sampling in this model is the placement of more UO_2 kernels than is physically accurate. Figure 4.1 demonstrates, as expected, that chord length sampling degrades with increasing matrix scattering.

Furthermore, the magnitude of the relative error in the predicted leakage fraction is also strongly a function of the total optical thickness of the interstitial absorber material. Even in very highly scattering conditions, chord length sampling shows very good agreement with benchmark results for spectrum average interstitial kernels. In this instance, even though additional, non-physical UO_2 kernels are likely to be sampled, the neutron is unlikely to undergo an interaction within these relatively optically thin regions. In other words, the geometry modeling error is of little consequence. Conversely, when the interstitial kernels are optically thick this modeling error is significant. Thus, while matrix scattering optical thickness governs the frequency of the geometry errors, the overall magnitude is governed largely by the optical thickness of the interstitial material.

A second set of parameterization data was compared for a slightly different heterogeneous mixture. Figure 4.3 presents results for a heterogeneous mixture

containing 600 UO_2 kernels of radius 1 mm. The material cross sections were scaled so as to give optical thickness parameters that matched those of Figure 4.1 for all 18 data points. Changes in the chord length sampling relative error are attributed to the increase in the kernel packing density. The chord length ratio for the new mixing statistics has been reduced from 15.2 to 9.8.

All cases show a degradation of chord length sampling accuracy in relation to Figure 4.1. In this more densely packed medium, geometry conflicts such as that illustrated in Figure 4.2 occur with greater frequency.

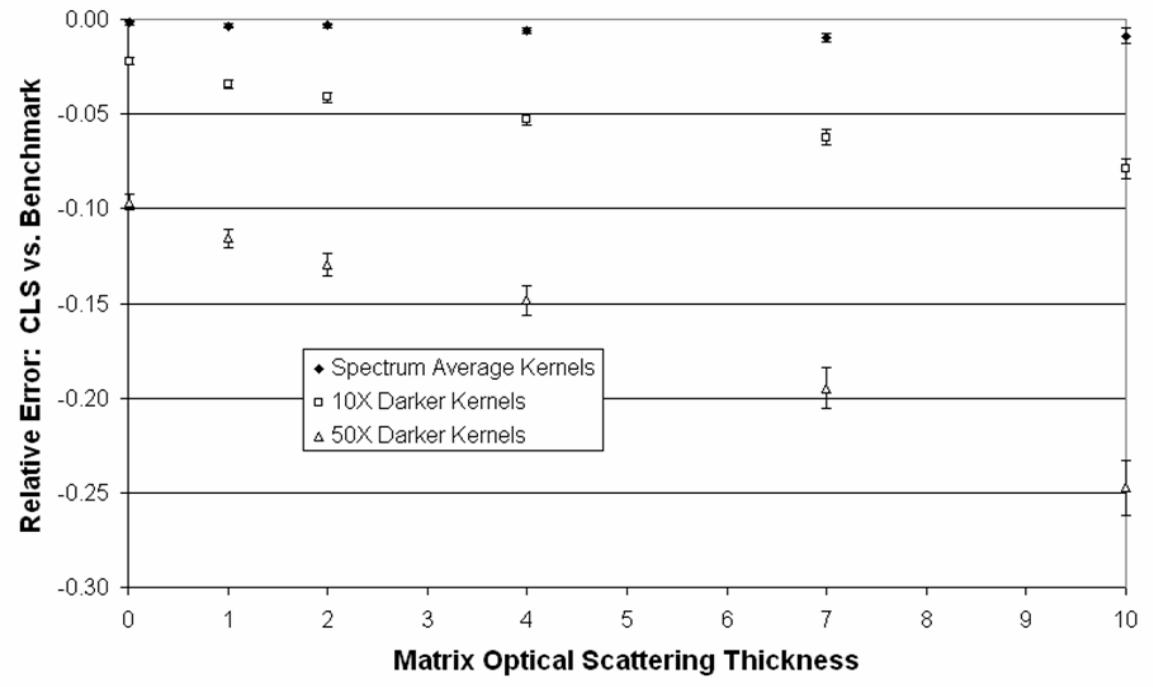


Figure 4.3: Relative Error of Chord Length Sampling Leakage Probability Denser Packing Fraction with 600 Interstitial UO_2 Kernels

As a final exercise for comparison we considered a mixture containing 400 UO_2 kernels but of radius 0.5 mm. For all 18 cases examined, chord length sampling shows agreement well within 1% of the benchmark method.

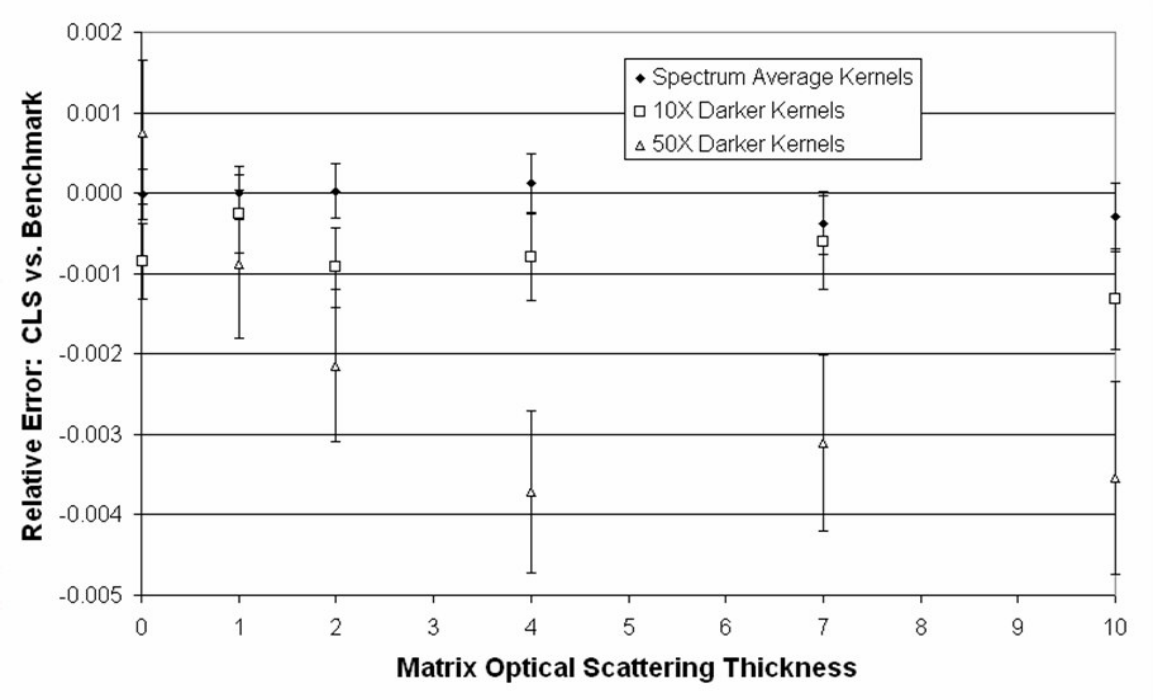


Figure 4.4: Relative Error of Chord Length Sampling Leakage Probability Sparse Packing with 400 0.5 mm Radius Interstitial UO_2 Kernels

The relative error in the ensemble average leakage probability is very small even for optically dark interstitial absorbers and a highly scattering matrix. Geometry modeling errors in three dimensions occur with much less frequency for a very sparsely packed medium.

In the LEUPRO pebble problem, the volume fraction of interstitial UO_2 kernels of radius 0.0251 cm is under 1%. Thus, the packing density of the optically thick kernels in this problem is quite sparse. Donovan's application of limited chord length sampling to

the LEUPRO benchmark showed very good agreement with published results. Thus, chord length sampling should be very capable of performing reasonably accurate simulations for most HTGR fuel analysis concerns.

Figure 4.5 is a compilation of results from all three geometry variations for the 10 times darker kernels. For very sparsely packed media (large mean chord length ratios) even under high scattering conditions, chord length sampling exhibits good agreement with the benchmark ensemble average leakage fractions. Accuracy degrades with both kernel packing density and with increasing matrix scattering.

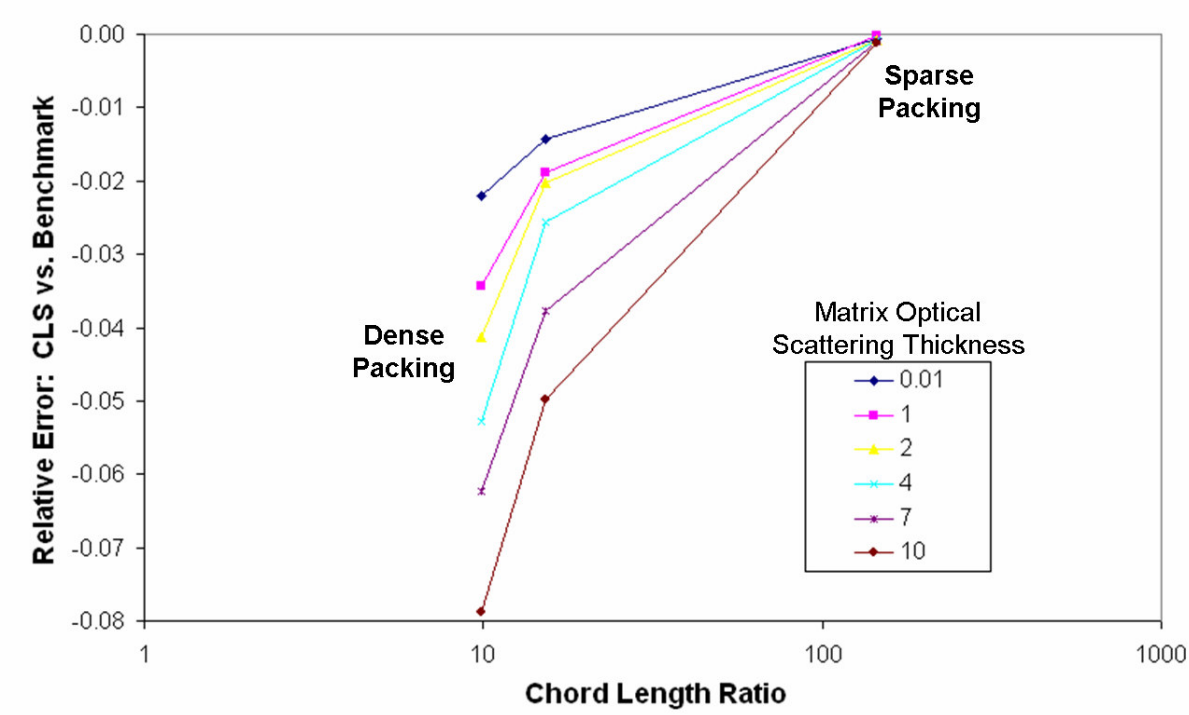


Figure 4.5: Variation of Chord Length Sampling Relative Error with Chord Length Ratio for 10X Spectrum Average Optically Dark Kernels

Figure 4.6 is a summary of error comparison results as function of chord length ratio and interstitial kernel optical thickness. All results were obtained using a uniform matrix optical scattering thickness of 1. These results indicate that chord length sampling

is highly accurate when applied to sparsely packed systems. Accuracy degrades with increased kernel packing density, increased matrix scattering, and increased kernel optical thickness.

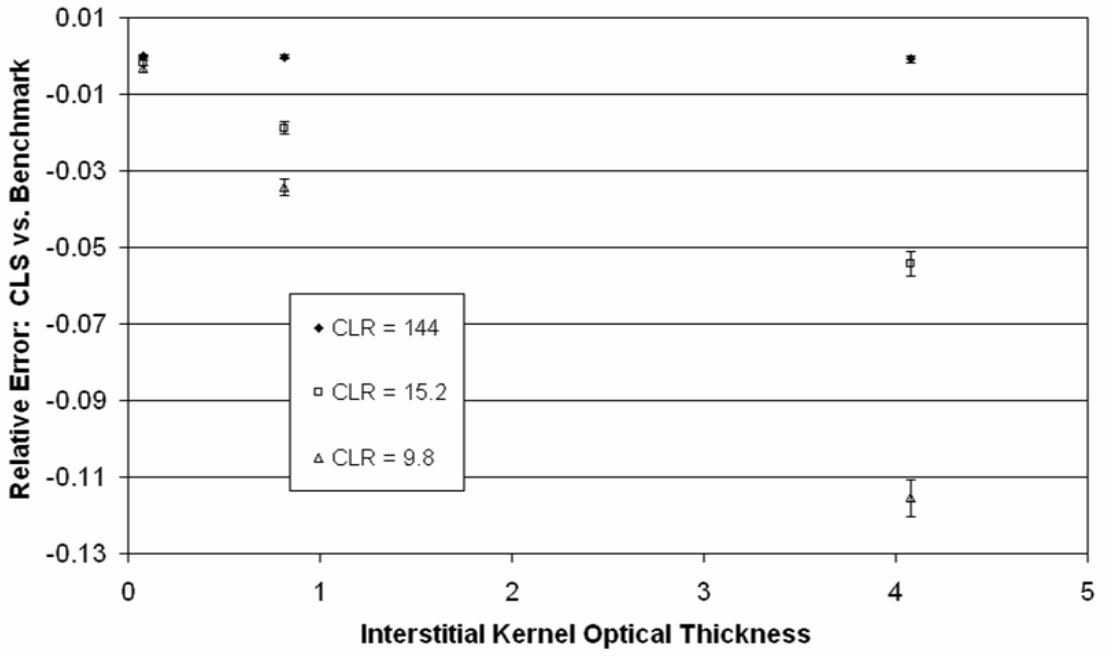


Figure 4.6: Variation of Chord Length Sampling Relative Error with Chord Length Ratio and Kernel Optical Thickness for Matrix Optical Scattering Thickness of 1.0

4.3 CONCLUSIONS

This parameterization has identified specific regimes in which the limited chord length sampling technique loses accuracy in three dimensional binary mixture transport simulations. Chord length sampling suffers most significantly in highly scattering problems involving dark, densely packed absorbers. It was shown that no single exclusive parameter such as optical scattering thickness is capable of predicting the accuracy of the chord length sampling method. By tabulating error trends as a function

of these non-dimensional parameters, these observations should be applicable to a wide array of random heterogeneous media transport applications.

It is important to note, however, that real neutron transport problems are not monoenergetic. The true continuous energy transport problem is a combination of these monoenergetic regimes. The optical thicknesses of all materials take on a wide range of energy-dependent values. For instance, in our 740 group energy structure the kernel optical thickness may vary by a factor of more than 1,500 between the optically thickest and optically thinnest energy groups. At most of these energies, the chord length sampling technique handles the transport very well. However, the vast majority of the loss in accuracy of the full continuous energy simulation is attributed to the certain specific parameter combinations identified above, namely highly scattering matrix conditions and dark, densely packed absorbers.

These results have helped to focus our research efforts on the advancement of two additional Monte Carlo algorithms. Chapter 5 discusses considerations for the implementation of an extended memory chord length sampling algorithm for reducing the frequency of geometry sampling conflicts. Chapter 6 describes the development of a hybrid benchmark and atomic mix algorithm for more efficiently representing optically dark absorbers in a highly scattering matrix.

Chapter 5: Implementation of an Extended Memory Chord Length Sampling Algorithm

This chapter describes the development of a chord length sampling routine that retains in memory the geometry regions a particle has traversed. The previous examination of chord length sampling errors trends confirmed a deficiency in the Levermore-Pomraning approximation's uncorrelated tracks assumption when the neutron has a high probability of scattering. Under such conditions, the frequency of events in which a neutron undergoes a scattering event and encounters a material region that is different from what it had previously traversed at the same location earlier in the history increases. Thus, chord length sampling allows for geometry interfaces to be placed in locations that would not be physically permissible. A chord length sampling algorithm that recollects a neutron's prior material traversals would preclude such geometry errors. The disadvantage of this strategy is that the increased fidelity of the geometry comes at the expense of additional computational effort.

In chord length sampling, the geometry interfaces between material regions within the heterogeneous mixture are assigned on-the-fly during the course of simulating each independent history. Each history begins without any established geometry. There are two mechanisms by which the geometry model is established via chord length sampling. Sampling kernel surfaces from the chord length PDF explicitly defines kernel locations, and sampling straight line flight segments between kernels implicitly defines matrix regions. Subsequently defining geometry that is in conflict with established matrix or kernel regions reduces the accuracy of the chord length sampling result. Thus,

both established kernel locations and established free-flight segments should be considered in a memory-based chord length sampling approach.

5.1 TRACKING ON KERNELS

The most straightforward algorithm revision for increasing a neutron's memory of previous encountered material regions is to record the (x, y, z) centroid coordinates of each spherical kernel. By retaining these coordinates, a neutron that traverses a kernel then leaves and has a backscatter collision within the matrix will have the possibility of encountering the same exact same kernel placement later in its history. Thus, the model will have a higher degree of accuracy in representing an acceptable physical arrangement. This additional fidelity comes at the expense of an additional distance calculation. For a neutron within the matrix, the modified simulation process will now require a total of four distances to be determined by:

i) calculating the distance to the pebble boundary, where leakage or reflection occurs,

ii) sampling the distance to a new material interface with chord length sampling,

iii) sampling the distance to a collision

iv) explicitly calculating the distance to all previously established kernel spheres.

If the distance to a previous kernel is the smallest of the four distances, the neutron's position is updated and the standard transport simulation within the previous kernel resumes. Item iv) is an identical calculation to those made using the benchmark method. Now however, we consider only the kernel spheres already established within the current kernel history. Thus, at the beginning of the history, the computational expense is quite low. Later in the history, for neutron histories that have survived many kernels without capture, this added computational requirement grows rapidly. The stored kernel

geometry is cleared when the neutron is captured or leaks from the system. This added computational expense will pale in comparison to that of the benchmark method for very large and complex heterogeneous mixtures. The LEUPRO benchmark algorithm, for instance, must consider all 9,394 kernels when assessing every matrix free-flight segment.

The second feature of tracking on kernels is to ensure that the spherical kernels do not overlap one another. If a new kernel is located with the chord length sampling procedure, before permanently fixing its (x, y, z) centroid coordinates, its location relative to all previously assigned kernels must be considered. The centroid of the new kernel is affixed in the same manner as in the standard chord length sampling routine: defining the interface point at which the neutron enters the sphere, sampling a kernel chord, and sampling the sense of the kernel chord relative to the kernel centroid. Before accepting this kernel location we must check its distance from all previously established kernels (x_i, y_i, z_i). In order to accept the new kernel coordinates $(x_{new}, y_{new}, z_{new})$ the following condition must be met.

$$(x_{new} - x_i)^2 + (y_{new} - y_i)^2 + (z_{new} - z_i)^2 > (2 \times rad)^2 \tag{5.1}$$

If overlap exists, the new kernel geometry is rejected. The neutron position is updated to the sampled interface position but the material designation remains as matrix and the simulation proceeds.

5.2 TRACKING ON TRACKS

Just as placement of new kernels must not conflict with established kernel locations, new kernel locations must also not interfere with previously established matrix regions. The matrix geometry is defined by the free-flight segments between matrix collisions and geometry interfaces. This strategy requires that all collision points be

recorded. An example of a geometry overlap conflict involving matrix segments is depicted below in Figure 5.1.

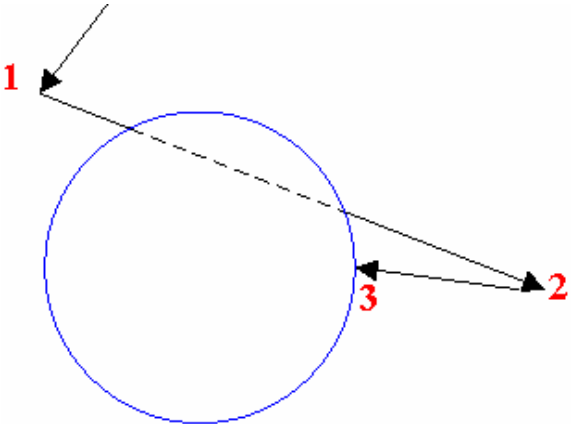


Figure 5.1: Example of Sampled Kernel Geometry Conflict with Previously Established Matrix Region

Figure 5.2 illustrates the geometrical relationship between a new chord length sampling sphere defined by its centroid C and an arbitrary matrix segment AB. The point P is the shortest distance from AB to C. Vector $\vec{r}$ defines point P relative to point A.

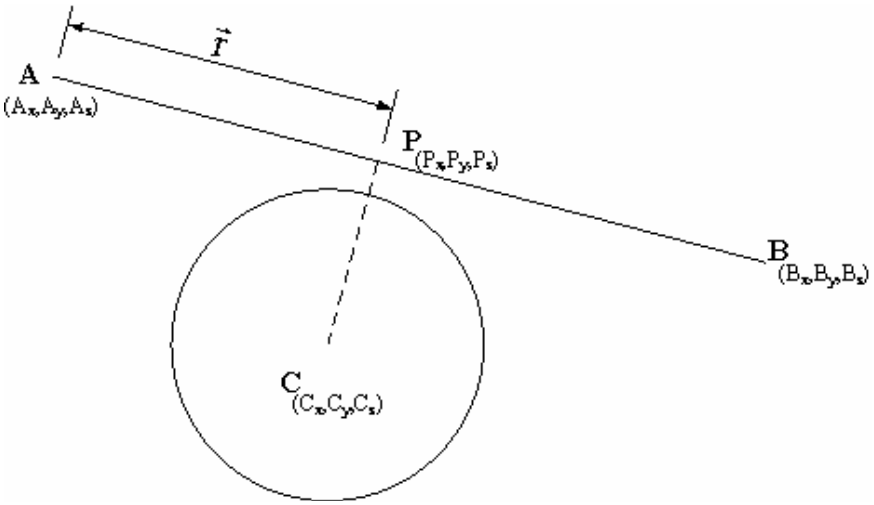


Figure 5.2: Geometry Considerations for Evaluating Track Overlap

$\|\vec{r}\|$ is the distance from point A to point P where

$$\|\vec{r}\| = \frac{AC \bullet AB}{\|AB\|^2} = \frac{(C_x - A_x)(B_x - A_x) + (C_y - A_y)(B_y - A_y) + (C_z - A_z)(B_z - A_z)}{\sqrt{(B_x - A_x)^2 + (B_y - A_y)^2 + (B_z - A_z)^2}} \tag{5.2}$$

The calculated value of $\|\vec{r}\|$ falls into one of five categories:

$\|\vec{r}\| = 0$ $P = A$

$\|\vec{r}\| = 1$ $P = B$

$\|\vec{r}\| < 0$ P lies on the backward extension of AB

$\|\vec{r}\| > 1$ P lies on the forward extension of AB

$0 < \|\vec{r}\| < 1$ P lies between A and B

A sphere overlapping the backward extension of a segment is illustrated below in Figure 5.3.

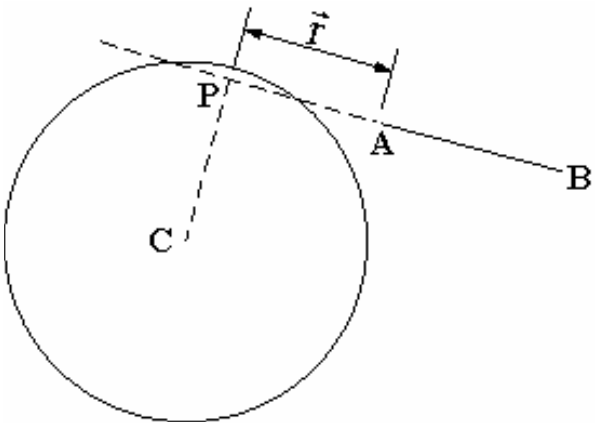


Figure 5.3: Geometry Check for Kernel Overlap of Finite Length Matrix Free-Flight Segments

Thus point P does not have to exist strictly between the end points of segment AB. In this case, the closest point between C and AB is the starting point of the segment, point A. Similarly, if $\|\vec{r}\| > 1$ the closest point between C and AB is point B. If $0 < \|\vec{r}\| < 1$ the value of P is determined according to

$$\begin{aligned} P_x &= A_x + \|\vec{r}\|(B_x - A_x) \\ P_y &= A_y + \|\vec{r}\|(B_y - A_y) \\ P_z &= A_z + \|\vec{r}\|(B_z - A_z) \end{aligned} \tag{5.3}$$

Then in order to confirm that the sphere will not overlap the segment the distance d between points C and the nearest point on segment AB is calculated. If d is less than the kernel radius the kernel location is rejected; otherwise the kernel location is acceptable and the transport simulation continues in normal fashion.

5.3 RESULTS AND DISCUSSION

The standard Monte Carlo algorithm was modified to incorporate the geometry tracking consideration described above. The logic flow of the extended memory chord length sampling algorithm is represented visually in Appendix B

The extended memory chord length sampling algorithm was applied to a random heterogeneous mixture similar to those discussed in Chapter 4. The model considered is a sphere of radius 2 cm containing 600 randomly arranged UO_2 kernels of radius 1.0 mm. This equates to a volume fraction of 7.5%. The matrix material is borated graphite and a vacuum boundary condition is applied at the pebble's outer surface. One group kernel cross sections were scaled to 50 times their nominal spectrum averaged values.

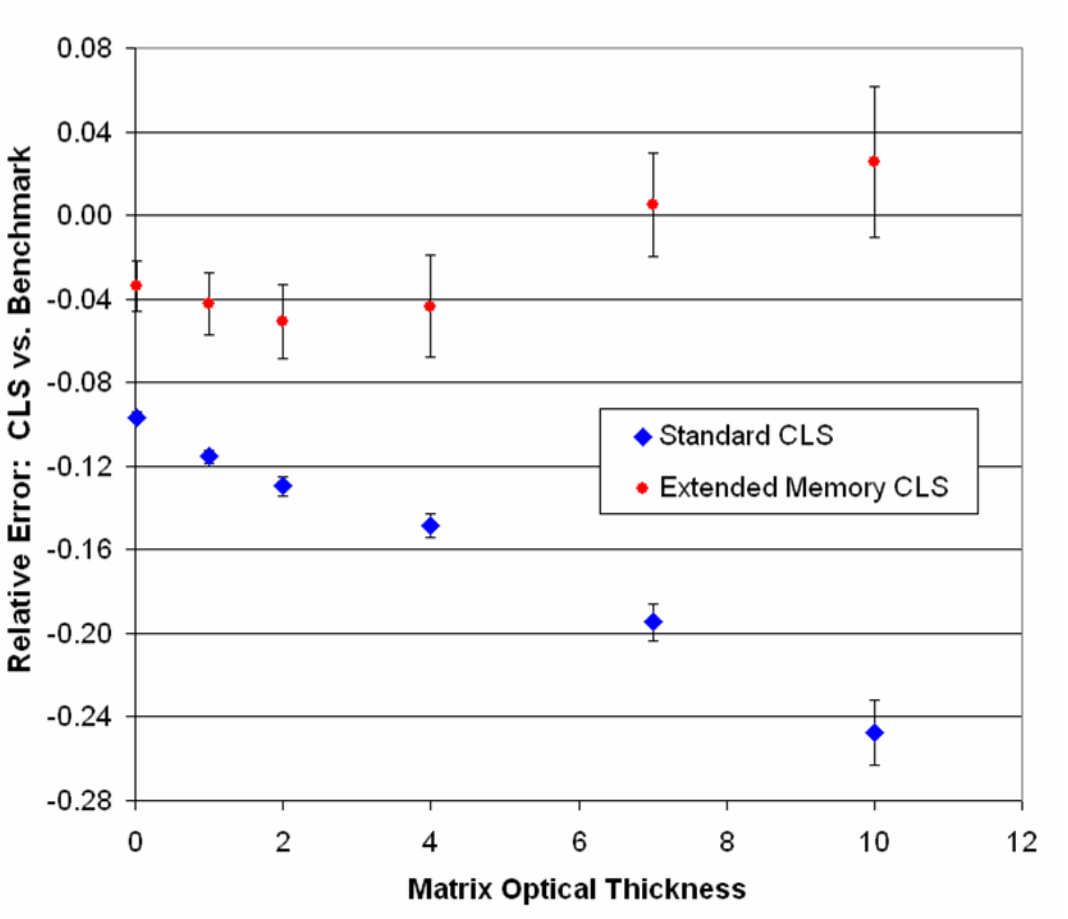


Figure 5.4: Comparison of Relative Errors for Standard and Extended Memory Chord Length Sampling in a 2 cm Pebble with 600 1.0 mm kernels in a Purely Scattering Matrix

As discussed previously, chord length sampling can yield an over-placement of UO_2 kernels for highly scattering matrix conditions. As the matrix scattering cross section is increased, the mean free path between scattering collisions is decreased and the frequency of scattering collisions increases. This means that the frequency of a neutron scattering backwards and seeing a new material increases. The net effect of this is over

predicting absorption by placing kernels in regions previously established as being matrix.

The extended memory chord length sampling algorithm shows improved agreement with the benchmark method for all cases. For the six cases presented in Figure 5.4 the total increase in computation time relative to the standard chord length sampling method is 21%. However, the total computational time of the extended memory chord length sampling is just 10% that of the explicit geometry delta tracking benchmark method. As illustrated in Figure 5.5 the increase in computation expense is much more considerable for thick, highly scattering matrix conditions than it is for near transparent matrix conditions.

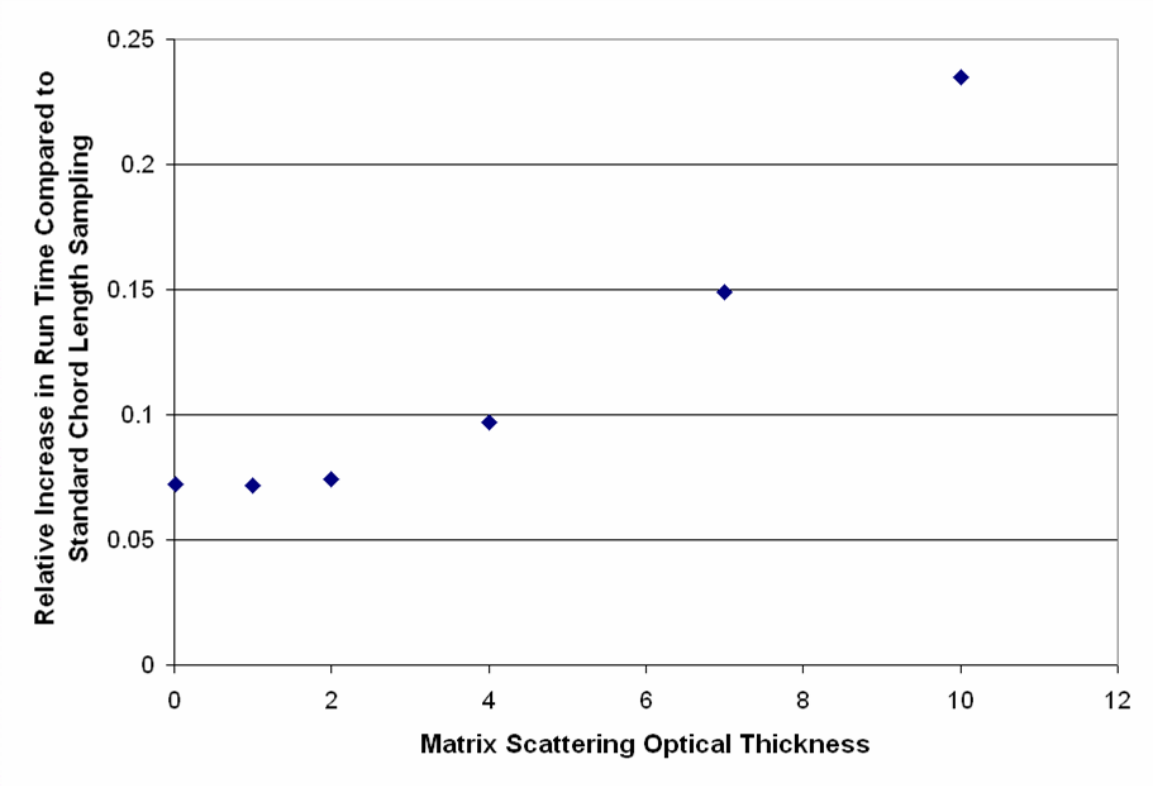


Figure 5.5: Relative Run Time Increase for Performing Extended Memory Chord Length Sampling for a Range of Matrix Scattering Conditions

An extended memory chord length sampling algorithm that records all previous kernel and matrix segments may be computationally prohibitive if applied to mixture geometries more intricate or expansive than that considered. For such cases, it may be permissible to maintain a record of only the neutron's recent history of materials traversed. This approach should retain the computational efficiency of chord length sampling while still precluding the effects of the most frequent geometry conflict errors. Further exploration of this topic is advised.

Chapter 6: Implementation of a Resonance Switch Algorithm

This chapter describes the development of a continuous energy Monte Carlo simulation technique for conducting neutron transport simulations within random heterogeneous materials. This approach uses a hybrid geometry model that draws from both the atomic mix and benchmark methods. This approach was chosen based upon the results of the chord length sampling error parameterization study discussed in Chapter 4. The breakdown of the uncorrelated tracks assumption is tied closely to the probability of interacting with an interstitial kernel. This probability of interaction is determined by the energy dependent total macroscopic cross section of the interstitial material.

Figure 6.1 below shows the potential inadequacy of simple homogenization in greater detail. A unit cell consisting of a single 0.0251 cm radius uranium dioxide kernel surrounded by graphite matrix was examined. A white reflective boundary condition was imposed as the edge of the spherical unit cell. This boundary condition replicates an isotropic neutron field. These results are consistent with similar calculations performed using MCNP (Ji *et al*., 2004) that demonstrated that low-enriched uranium dioxide TRISO fuel particles, despite their very small diameter, will cause a localized flux depression for resonance energy neutrons. Thus, the uranium kernel appears to be optically thick to neutrons in the 6.57 eV-6.77 eV resonance energy range but optically thin to neutrons at other energies. Although this energy range comprises just a small fraction of the total flux, resonance capture effects play an important role in the neutron chain reaction. The atomic mix homogenization method is inadequate for predicting the neutronic behavior in this particular phase space region.

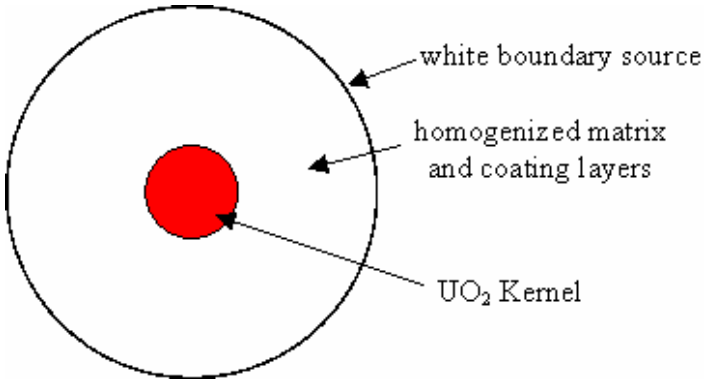


Figure 6.1: UO_2 Kernel Unit Cell

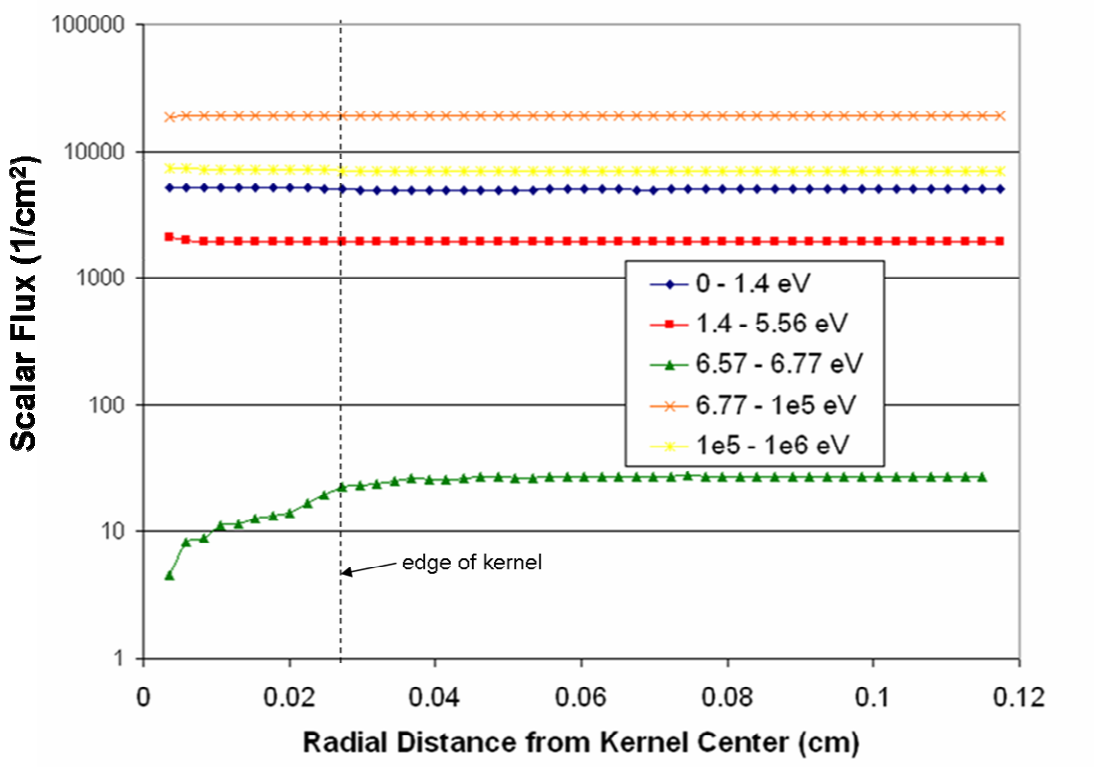


Figure 6.2: Kernel Radial Flux Profile

Prinja and Fichtl (2005) made a similar observation using deterministic techniques. They noted that when obtaining iterative solutions to the Levermore-Pomraning equations, convergence rates degraded within optically thin materials. They demonstrated that the coupled Levermore-Pomraning equations reduced to the atomic mix model when all materials were optically small compared to the neutron mean free paths. They demonstrated that the atomic mix model could provide a good approximation to the Levermore-Pomraning equation for very optically thin regions and using this also devised an atomic mix synthetic acceleration scheme for deterministic calculations in binary statistical mixtures.

The total microscopic cross section for ^{235}U as a function of energy is plotted in Figure 6.3. The resonance region of energies ranging from a few eV to a few hundred eV is the focus of our investigation. In this energy range the cross sections exhibit many sharply peaked maxima. At these discrete energies, the sum of the compound nucleus center of mass energy and the neutron binding energy matches one of the excited quantum energy levels of the compound nucleus. A neutron existing at the same energy as one of these resonance cross sections is highly likely to undergo an interaction. All nuclei except ^{1}H exhibit this resonance behavior (Knief, 1992).

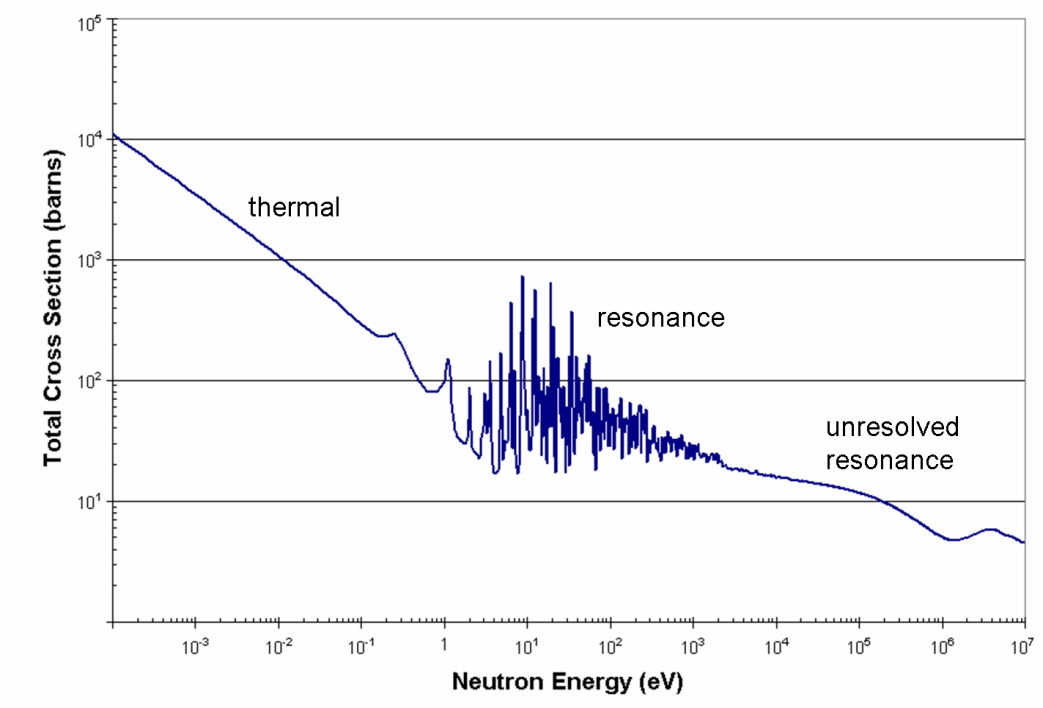


Figure 6.3: ^{235}U Total Cross Microscopic Cross Section

The most significant resonance peaks in uranium-fueled reactors and their respective microscopic cross sections are listed below in Table 6.1.

Table 6.1. Low Lying ^{238}U Resonance Cross Sections (Duderstadt and Martin, 1976)

Neutron Energy (eV)	Total Microscopic Cross Section σ (b)
6.67	2.16×10^{5}
20.90	3.19×10^{4}
36.80	3.98×10^{4}
66.54	2.14×10^{4}
102.47	1.86×10^{4}
116.85	1.30×10^{4}
165.27	2.41×10^{3}
208.46	8.86×10^{3}

The prevalence of these resonance peaks is more easily noted in the cross section plot presented on a linear scale in Figure 6.4 below. This data has been grouped according to our 740 group energy structure. In general, the lower the energy, the broader the resonance and the more effective it is in absorbing neutrons (Duderstadt and Martin, 1976). Therefore, the most significant of all ^{238}U resonance absorption reactions is the lowest lying resonance peak at 6.67 eV.

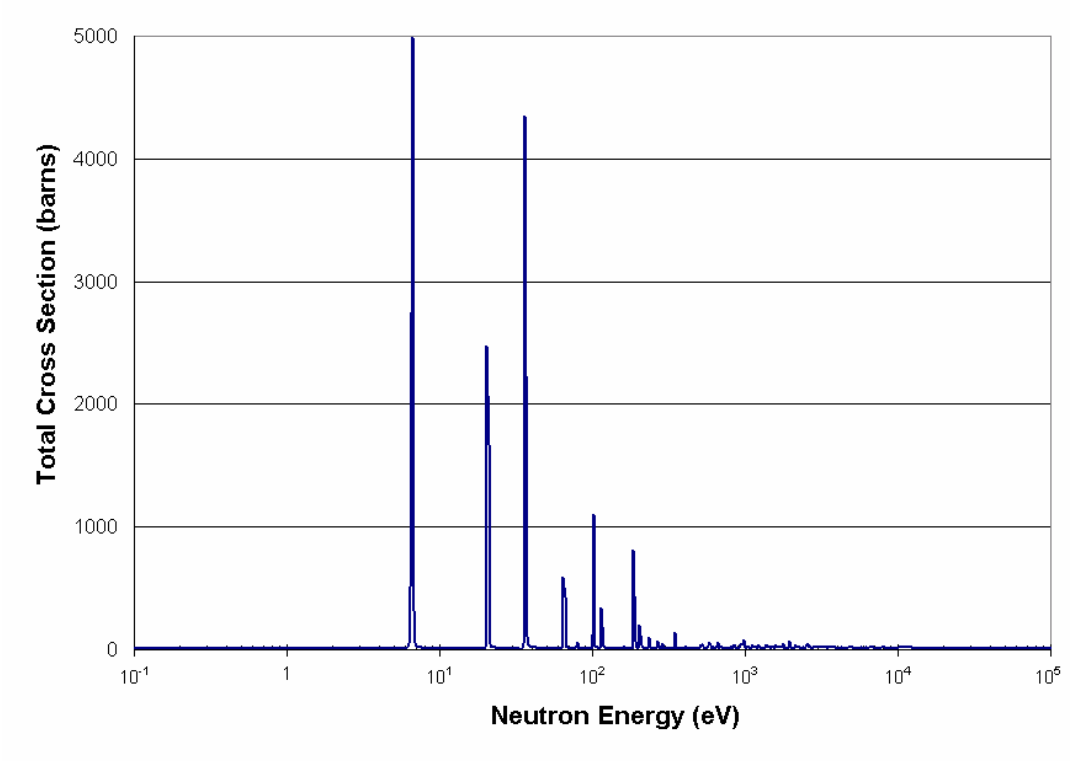


Figure 6.4: ^{238}U Total Microscopic Cross Section Illustrating Dominance of Low-Lying Resonance Peaks

The absorption of neutrons within resonance peaks while slowing down from fission energies is one of the primary loss mechanisms in nuclear reactors. Resonance absorbers such as ^{238}U play a major role in determining the reactor criticality. An illustrative example is a reactor fueled by natural uranium. If the reactor consists of a uniform mixture of natural uranium and graphite or heavy water moderator, too many neutrons will be absorbed by the ^{238}U resonances. Conversely, if the same quantities of natural uranium are lumped into rods or bundles, a critical chain reaction can be sustained. Thus, it is important to model the detailed geometry corresponding to these

resonance reactions. Graphite does not have any large resonances. Thus, our efforts here are focused solely capturing the resonance behavior of the UO_2 kernels.

6.1 SINGLE RESONANCE BAND METHOD

The single resonance band method model partitions the continuous energy Monte Carlo simulation into two geometry models. A fully explicit detailed geometry model is used in tracking neutrons whose energies fall within some resonance region. Transport simulations of neutrons at all other energies use a homogeneous atomic mix model. Initially, three energy bands were tested: 6 eV-7eV, 6 eV-40 eV, and 6 eV-210 eV. The first band encompasses just the 6.67 eV resonance. The second band contains the 6.67 eV, 20.90 eV, and 36.80 resonances, and the third and largest resonance band includes all eight resonance peaks listed in Table 6.1.

After each change in the neutron's energy, a comparison is made to determine whether the neutron energy is within the limits of the resonance band. If the particle's energy is within a resonance region the detailed geometry model is used; otherwise, the simulation is performed assuming a homogeneous region. For a neutron just entering the resonance band, before switching to explicit geometry tracking, we must first determine the material at the neutron's current location. This same material determination is inherent to the processing of collisions using delta tracking. The resonance switch technique was readily implemented by modifying the benchmark delta tracking algorithm.

This single resonance band technique was tested using a modified form of the LEUPRO k_∞ benchmark problem. The coating layers surrounding each UO_2 kernel were omitted as was the outermost moderator shell surrounding the pebble. Figure 6.5 shows the convergence of the k_∞ calculation using each of the selected resonance bands and also

the results of the benchmark and atomic mix methods. As expected, the more of the resonance region included in the explicit geometry model, the better the agreement with the benchmark technique.

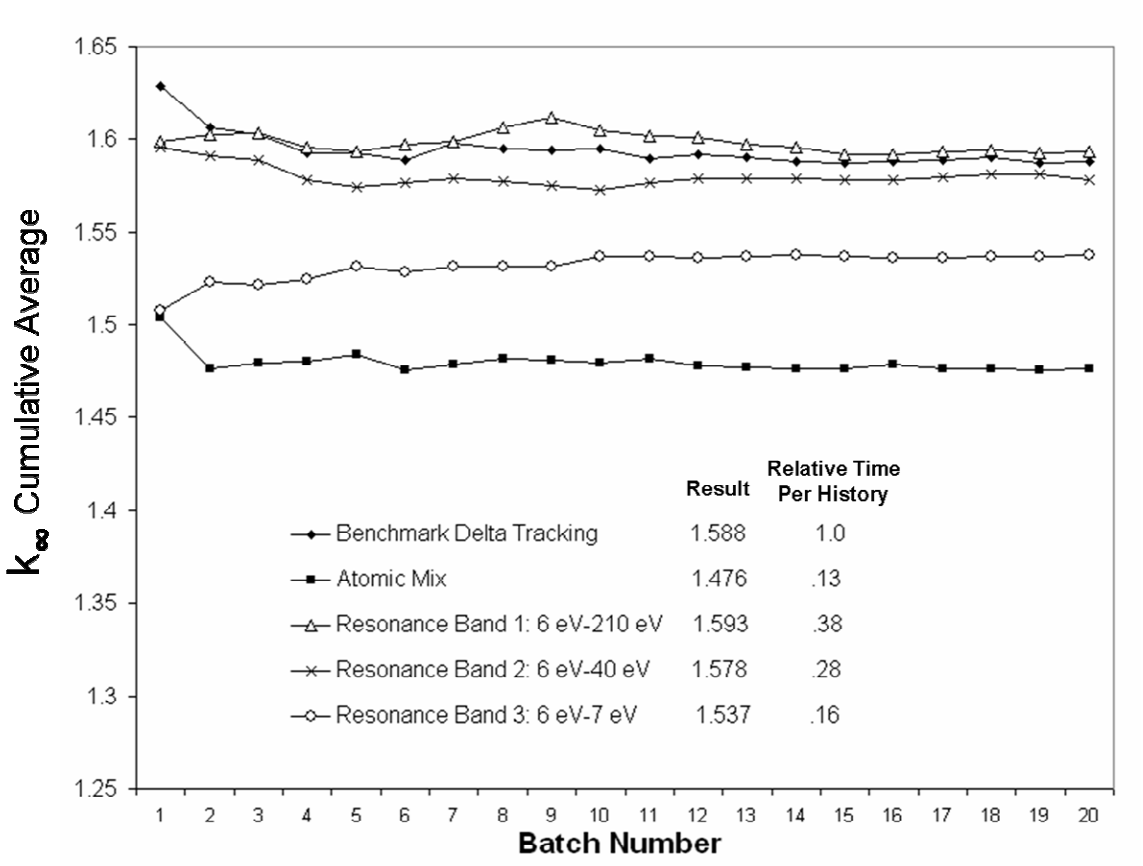


Figure 6.5: Convergence of k_∞ Simulation for Single Band Resonance Switch Method

The relative error of the fully homogeneous atomic mix model is reduced by almost 60% simply by modeling the geometry in the vicinity of the 6.67 eV resonance explicitly. Simply by focusing the computational effort and rigorously tracking through the explicit geometry model only if the neutron energy lies near this one localized resonance, the accuracy of the homogeneous model is substantially improved.

Furthermore, the computation time of the 6 eV-7 eV resonance band method is only 16.5% that of benchmark method using delta tracking. When this resonance band is expanded to envelop all 8 low-lying ^{238}U resonances, the error due to the homogeneous model is decreased by more than 95%. The 6 eV-210 eV resonance band result agrees to within 0.4% of the benchmark method.

6.2 OPTICAL THICKNESS THRESHOLD METHOD

The 6 eV-210 eV resonance band exhibits agreement to within 0.4% of the benchmark result. A total of 156 out of 740 energy groups are flagged as resonance groups for inclusion in the explicit geometry model. However, only a handful of these groups actually contain resonance peaks. Rather than imposing rigid upper and lower energy boundaries for the resonance treatment, we flag all energy groups above thermal energies as being resonance groups if the group optical thickness of the UO_2 kernel exceeds some threshold value. The aim of this effort is to further reduce the computational expense associated with unnecessarily modeling the heterogeneous mixture geometry within optically thin energy groups.

The optical thickness of the kernel is a function of the total macroscopic cross section and the radius of the kernel. Before the beginning of the simulation, an initial algorithm is executed to sweep through the group structure and flag all energy groups whose interstitial kernel optical thickness exceeds some threshold value. After every change in the neutron's energy, a check is performed to determine if the new energy falls into one of these optically thick resonance groups. Figure 6.6 illustrates the selected resonance groups for an optical thickness threshold of 0.2.

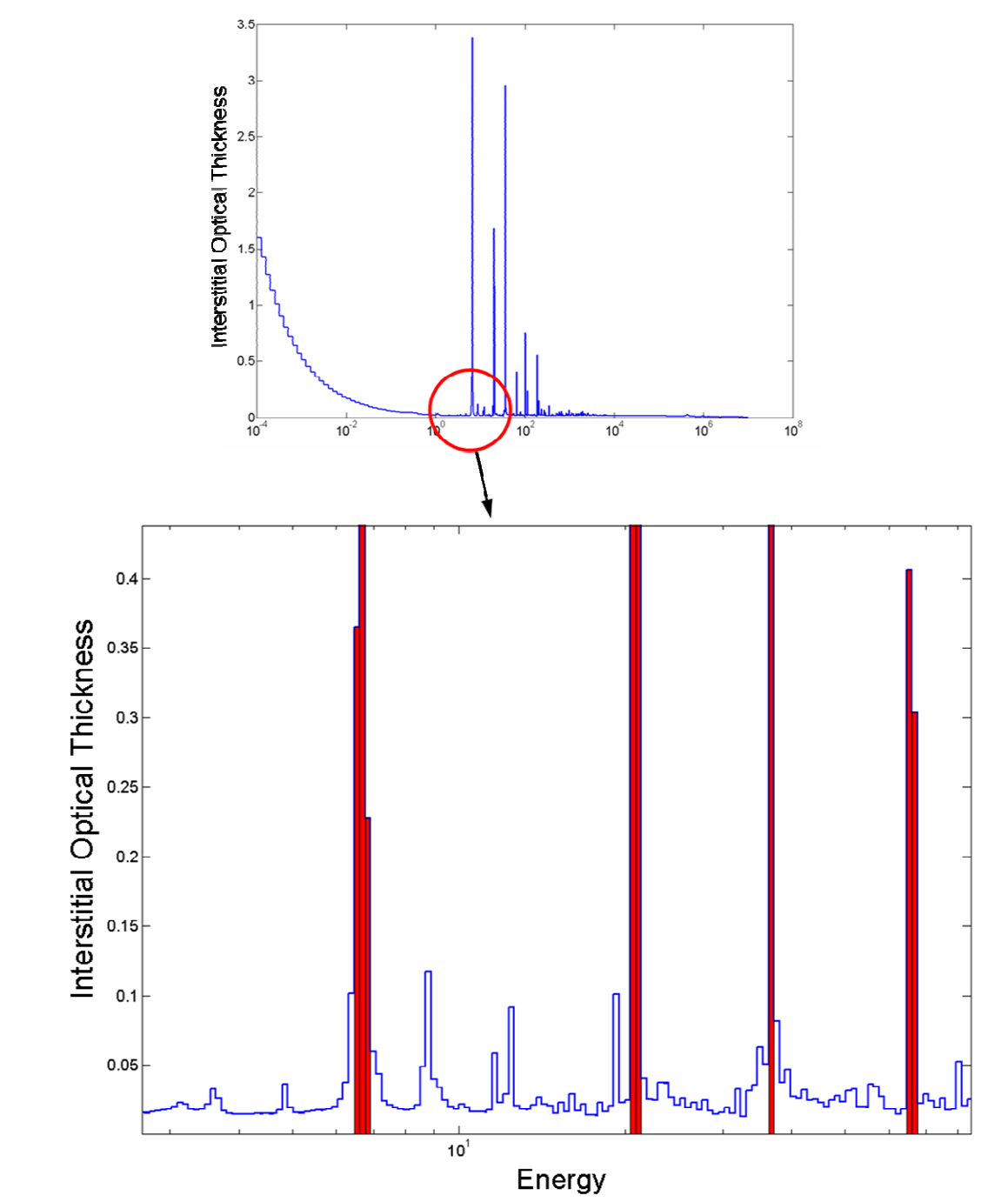


Figure 6.6: Interstitial Kernel Dark Resonance Groups for an Optical Thickness Threshold of 0.2

The thermal range energy groups having optical thickness exceeding the threshold produced negligible effects on k_∞ and thus were omitted from the threshold method. This strategy was tested on the LEUPRO k_∞ problem using optical thickness threshold values of 3, 1, and 0.2. These cutoffs were selected so as to represent the same grouping of resonance peaks as in the previous single band resonance method evaluation. A comparison of results from the single resonance band method and optical thickness threshold method is given below in Table 6.2.

Table 6.2. Comparison of Single Resonance Band and Optical Thickness Threshold Results

Method	k_∞ Relative Error	Number of Resonance Groups	Relative Time per History
Single Band 6 eV—210 eV	0.0030	156	1.00
Single Band 6 eV—40 eV	0.0064	84	0.80
Single Band 6 eV—7 eV	0.0320	8	0.50
Optical Thickness Threshold 0.2	0.0309	11	0.67
Optical Thickness Threshold 1.0	0.0603	4	0.53
Optical Thickness Threshold 3.0	0.0748	1	0.45
Atomic Mix	0.0706	0	0.42

The optical thickness threshold is successful in decreasing the number of resonance groups over which detailed geometry tracking must be performed. There is however, a notable loss of accuracy relative to the single resonance band method.

6.3 PEAK BROADENED OPTICAL THICKNESS THRESHOLD METHOD

Figure 6.7 illustrates the assignment of resonance groups using an optical thickness threshold of 3. Although the cutoff thickness identifies the magnitude of the peak, the cutoff parameter doesn't fully capture the broadness of the 6.67 eV resonance.

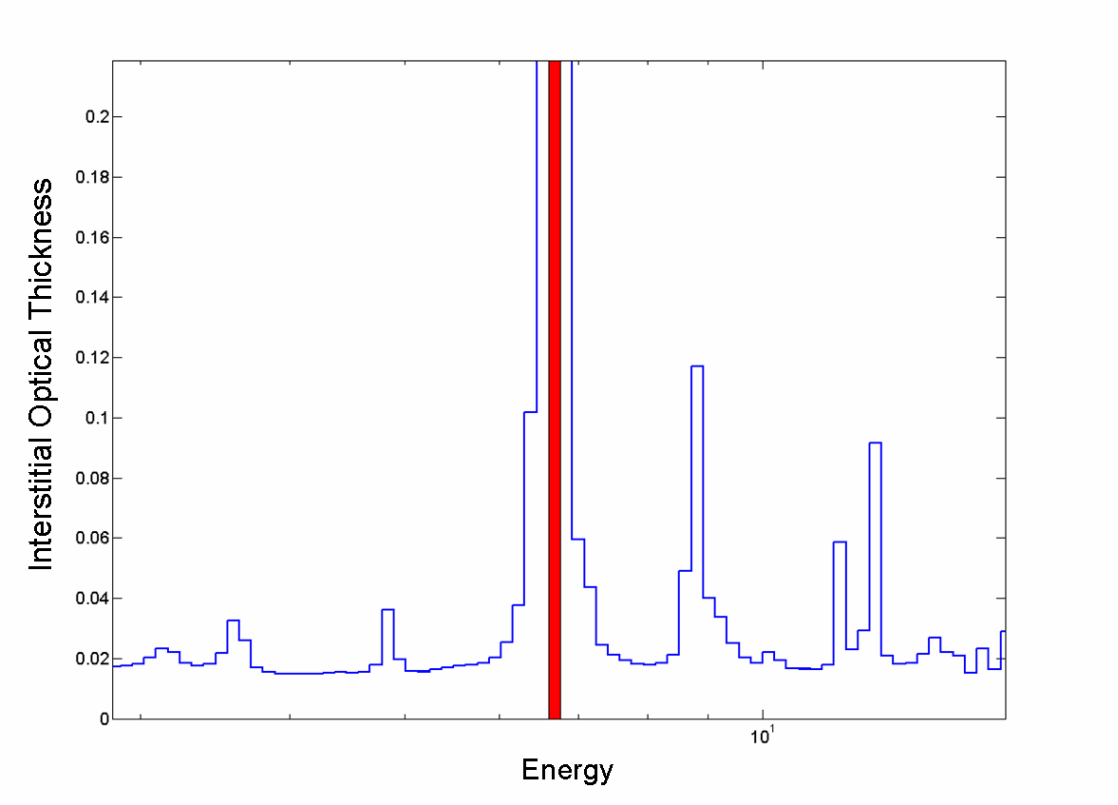


Figure 6.7: Interstitial Kernel Dark Resonance Groups for an Optical Thickness Threshold of 3.0

Table 6.3 lists the macroscopic total, absorption, and elastic scattering cross sections within the energy group containing the 6.67 eV resonance peak.

Table 6.3. Kernel Macroscopic Cross Sections for 6.67 eV Resonance Group

Cross Section [b]	^{238}U	^{235}U	^{16}O
Total	100.708	0.115	0.189
Absorption	94.306	0.065	5.77e-7
Elastic Scattering	6.402	0.050	0.189

Approximately, 99.7% of all neutrons interactions within the UO_2 kernels occur with the ^{238}U isotope. Of those collisions, 6.35% are scattering events. These scattering events may cause the neutron energy to drop out of the single optically thick resonance group. In reality, the energy loss from such a scattering event is small, and the neutron would likely still be absorbed within the tail of the resonance peak.

The maximum fractional energy loss, α, in an elastic scattering collision is defined by

$$E'_{\min} = \alpha E = \left(\frac{A-1}{A+1}\right)^2 E \tag{6.1}$$

where E is the incident neutron energy, $E'_{\min}$ is the minimum outgoing neutron energy and A is the atomic mass of the target nuclide. For $A = 238$, $E'_{\min} = 0.983E$. In lethargy space this amounts to

$$\Delta u_{\max} = \ln E'_{\min} / E = \ln 1 / 0.983 = 0.0171 \tag{6.2}$$

The energy group structure in this region is divided into 100 groups per decade of lethargy. Thus,

$$\Delta u_{group} = \ln 10 / 100 = 0.023 \tag{6.3}$$

The maximum neutron energy loss due to a scattering collision is less than the width of a single energy group. Therefore, the neutron can scatter downward at most only one group due to each scattering collision. The optical thickness threshold

sweeping algorithm was amended to broaden each of the resonance peaks by an additional one group to take into account neutrons that slow down to energies just above the resonance energy either within, or in the vicinity of, a fuel particle. These neutrons are likely to still remain in the vicinity of a fuel particle the next time they undergo an energy loss; therefore the peak broadened approach explicitly models the spatial distribution of neutrons scattering *into* resonance energies as well. Figure 6.8 depicts a portion of the UO_2 kernel macroscopic cross section with resonance groups obtained this peak broadening approach. The value kernel optical thickness threshold value used was 0.2.

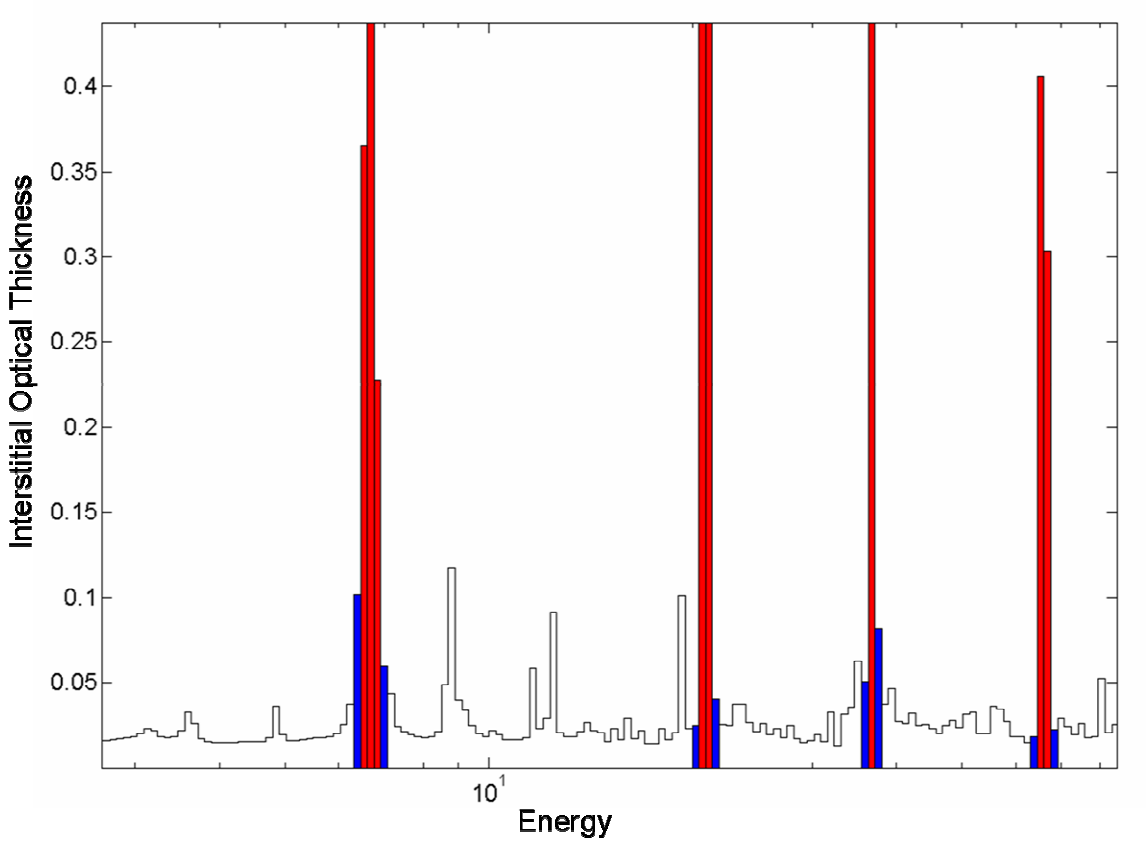


Figure 6.8: Peak Broadened Dark Resonance Groups for an Optical Thickness Threshold of 0.2

The peak maxima that exceed the optical thickness threshold are darkened in red. Those groups flagged as optically thick by the one-group peak broadening method are shown in blue. A comparison of k_∞ results from the original optical thickness threshold method and the new peak broadened method is given below in Table 6.4

Table 6.4. Comparison of Standard and Broadened Optical Thickness Threshold Results

Method	k_∞ Relative Error	Number of Resonance Groups	Relative Time per History
Narrow Optical Thickness 3.0	0.0748	1	0.45
Narrow Optical Thickness 1.0	0.0603	4	0.53
Narrow Optical Thickness 0.2	0.0309	11	0.67
Broadened Optical Thickness 3.0	0.0379	3	0.53
Broadened Optical Thickness 1.0	0.0021	10	0.74
Broadened Optical Thickness 0.2	0.0015	25	0.86

The peak broadened optical thickness threshold approach shows significant increases in accuracy for little added computational expense. Thus, the peak broadened optical thickness threshold approach is the most optimal of the resonance switch methodologies examined. This strategy is more accurate than the original optical thickness threshold method and avoids unnecessarily tracking particles through the optically thin valleys of the resonance range.

Figures 6.9 and 6.10 present a final comparison of relative errors and run times associated with all resonance switch methods considered.

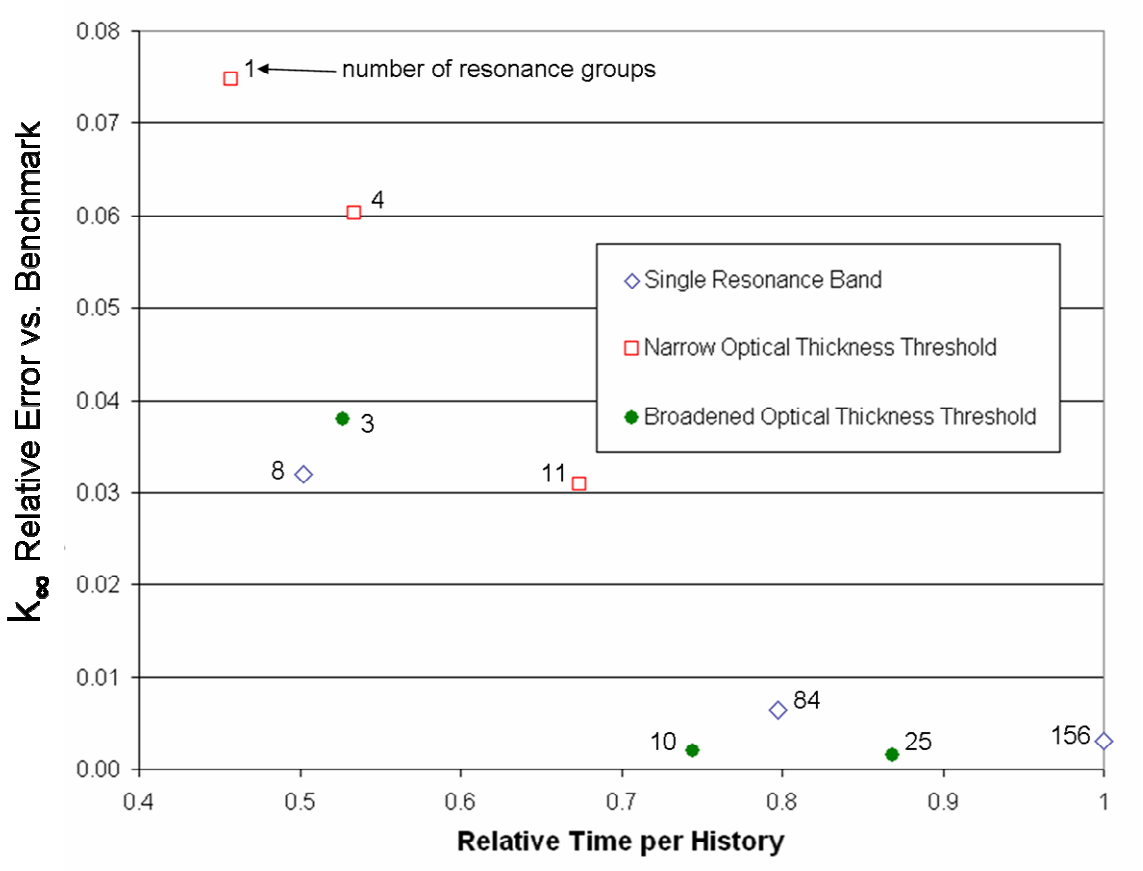


Figure 6.9: Results Summary for Optical Thickness Threshold Resonance Switch Method Relative Errors for k_∞ Simulation

The expected general trend is observed. In general, as the number of optically thick resonance groups is increased, the k_∞ result calculated by the resonance switch method approaches that of the benchmark explicit geometry model. As was observed in the peak broadened threshold comparison given in Table 6.4, a higher optical thickness threshold cutoff with peak broadening may produce a more accurate simulation than a lower optical thickness threshold that considers only the groups corresponding to the peak maxima.

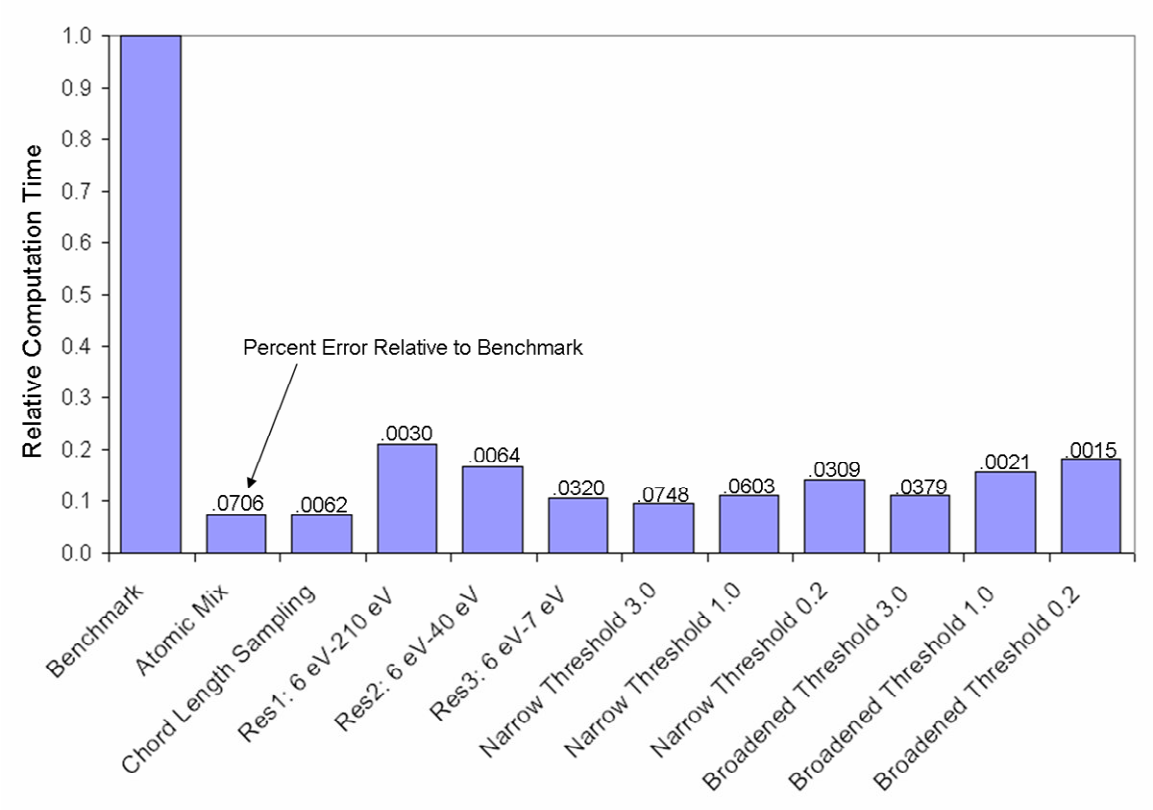


Figure 6.10: Relative Timing Comparison of Monte Carlo Simulation Algorithms

Thus far, the value of the optical thickness threshold parameter has been selected specifically to capture the low lying resonance of ^{238}U. A more systematic and versatile approach to the selection of a suitable optical thickness threshold was desired. For this purpose we considered an arbitrary spherical absorber and imposed a white boundary source to replicate an infinite uniform monoenergetic neutron flux field. The number of neutrons reaching the center of the absorber will be less than that which is incident at the outer boundary. The spatial self-shielding factor, f, was calculated assuming that all collisions within the sphere result in absorptions. f is the ratio of the average scalar flux within the absorber to the incident scalar flux at the absorber boundary.

$$f = \bar{\phi}_{absorber} \Big/ \phi_{incident} \tag{6.4}$$

Results from this non-dimensional analysis are plotted in Figure 6.10.

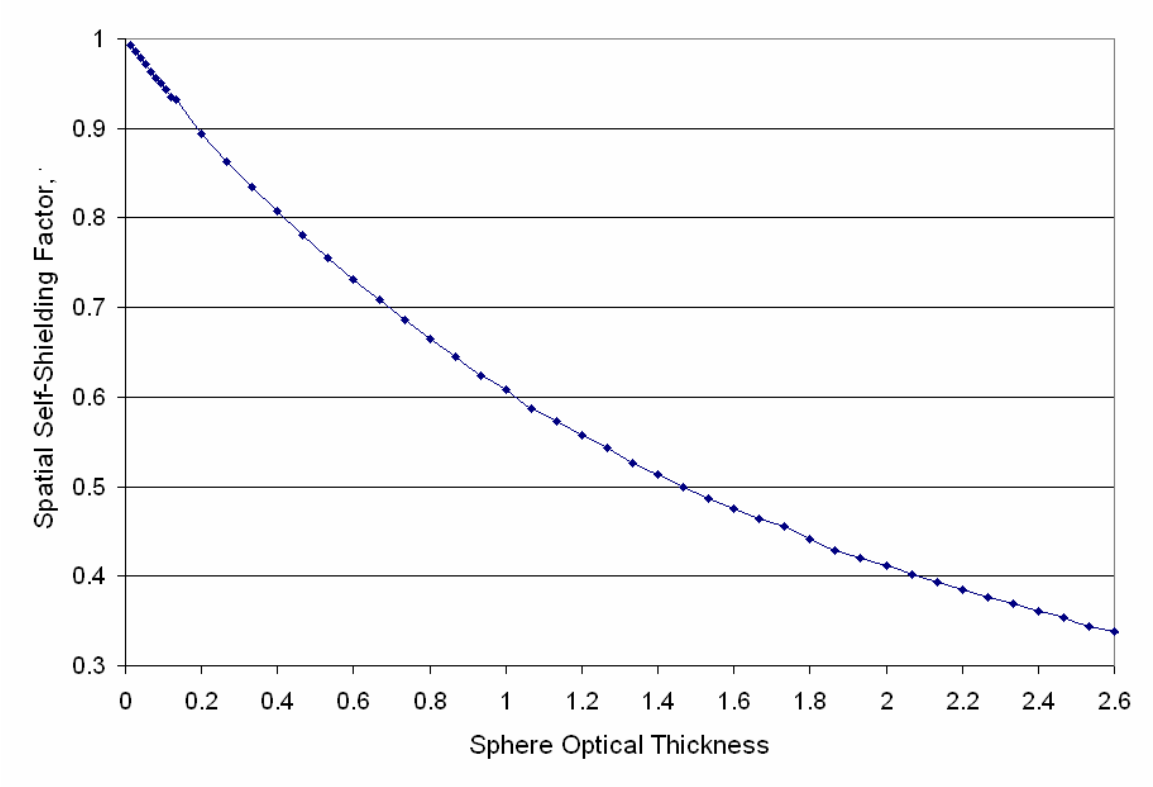


Figure 6.11: Spatial Self-Shielding Factor for a Spherical Absorber subject to a White Boundary Source

This result is consistent with similar empirical results obtained by Hurwitz and Zweifel (1956). For an optical thickness of 0.1 the spatial self-shielding factor f of the spherical absorber has a value of approximately 0.95. This value was assumed in calculating making a final comparison of methods for the full LEUPRO benchmark problems. The total number of energy groups exceeding this optical thickness threshold is 38. Again the four thin coating layers surrounding the UO_2 TRISO kernels have been

homogenized with the surrounding graphite matrix. The remaining problem specifications are consistent with the values given in Table 3.1. Final results for the full pebble LEUPRO benchmark are summarized in Table 6.5.

Table 6.5. Results Summary for k_∞ Calculation

Method	k_∞	Relative Error	Relative Time Per History
Benchmark (Delta)	1.7192	-	1.0
Atomic Mix	1.6138	0.0613	0.21
Standard Chord Length Sampling	1.7189	0.00017	0.20
Resonance Switch	1.7197	0.00029	0.44

The resonance switch method shows excellent agreement with our benchmark result. Both the chord length sampling method and the resonance switch method produce results that lie very near that of the benchmark calculation. The chord length sampling approach does require less computation time. The drawback of the chord length sampling approach is that it requires a mean chord length parameter to define the binary mixture. The relative calculation time of the chord length sampling results presented above does not consider this initial overhead to empirically obtain a mean matrix chord length parameter. For uniform non-overlapping spheres packed within a spherical shell, the mean chord length can be empirically determined in an efficient manner (Donovan, 2003c), but the cost of this computation may rise substantially for more complex mixture geometries. In contrast, the resonance switch method is fully extensible to all heterogeneous fuel assembly geometries. This method should be beneficial in

performing Monte Carlo simulations within any heterogeneous mixture in which one or more of the constituent materials are optically thick within some appreciable fraction of the energy spectrum. The focus here has been on the treatment of fissile uranium resonances. However, this method is not strictly limited to uranium based fuels. The resonance switch method could easily be applied to plutonium, thorium, or mixed-oxide HTGR fuel elements in which accurate treatment of resonance behavior is of equal importance.

Chapter 7: Conclusions

Neutron transport calculations for the random heterogeneous mixtures of interest in nuclear engineering applications are mathematically complex and computationally burdensome. A framework of Monte Carlo computer codes was generated to examine the relative costs and benefits of benchmark, atomic mix, and chord length sampling solution techniques. This work was performed in order to improve our understanding of radiation transport in random heterogeneous mixtures and identify potential areas in which further algorithm improvements may be warranted.

The accuracy of chord length sampling relative to the benchmark method was examined under a variety of conditions and tabulated as a function of three non-dimensional parameters: interstitial optical thickness, matrix scattering optical thickness, and chord length ratio. At any given neutron energy, any binary heterogeneous media neutron transport problem can be uniquely characterized by these three parameters. Error trends observed here as a function of these parameters should be extensible to other material and geometry combinations.

As expected, chord length sampling loses accuracy when the matrix optical scattering thickness is increased. This work demonstrated that the consequences of geometry conflict errors that arise subsequent to these matrix scattering events are inconsequential except in the presence of optically thick interstitial absorbers. The degradation of chord length sampling was shown to be more severe for higher kernel packing densities. In more densely packed geometries, geometry conflict errors occur with greater frequency. For a very sparse packing of interstitial kernels, chord length sampling performed very well under all conditions.

This dimensionless parameter set can be used to dictate a best method of solution. But because the optical thickness parameters vary as a function of the energy, the best solution method will also vary as a function of energy. This identification revealed the potential for a wide array of potential hybrid approaches for handling the energy dependent transport process. The resonance switch method is one such hybrid approach combining of benchmark and atomic mix algorithms, but there is a host of other similar methodologies that may be well suited to address random heterogeneous media transport problems. This may include a hybridized atomic mix and chord length sampling algorithm. With the initial success of the resonance switch treatment, further investigation of such energy optimized hybrid approaches is warranted.

New contributions of this research include the development of two new methods for performing Monte Carlo transport simulations in random heterogeneous mixtures. Extended memory chord length sampling produced a considerable increase in accuracy when applied to a mixture with a dense packing of optically thick interstitial kernels. The increase in accuracy of an extended memory chord length sampling algorithm over standard chord length sampling had not previously been demonstrated for a three-dimensional random mixture. However, under highly scattering matrix conditions, the increase in computational expense was shown to be considerable. In some cases, extended memory chord length sampling required run times increases of more than 50% above those of standard chord length sampling.

A hybrid algorithm that combines homogeneous and explicit geometry models within a single continuous energy Monte Carlo simulation was also investigated. This approach restricted the computationally expensive explicit geometry tracking to be performed only within the optically thick resonance peaks. A significant contribution of this work was in demonstrating that by partitioning the geometry representation within a

single Monte Carlo simulation into homogenous and heterogeneous energy-dependent models, acceptable ensemble average results can be obtained in a fraction of the run time of the detailed explicit geometry benchmark method.

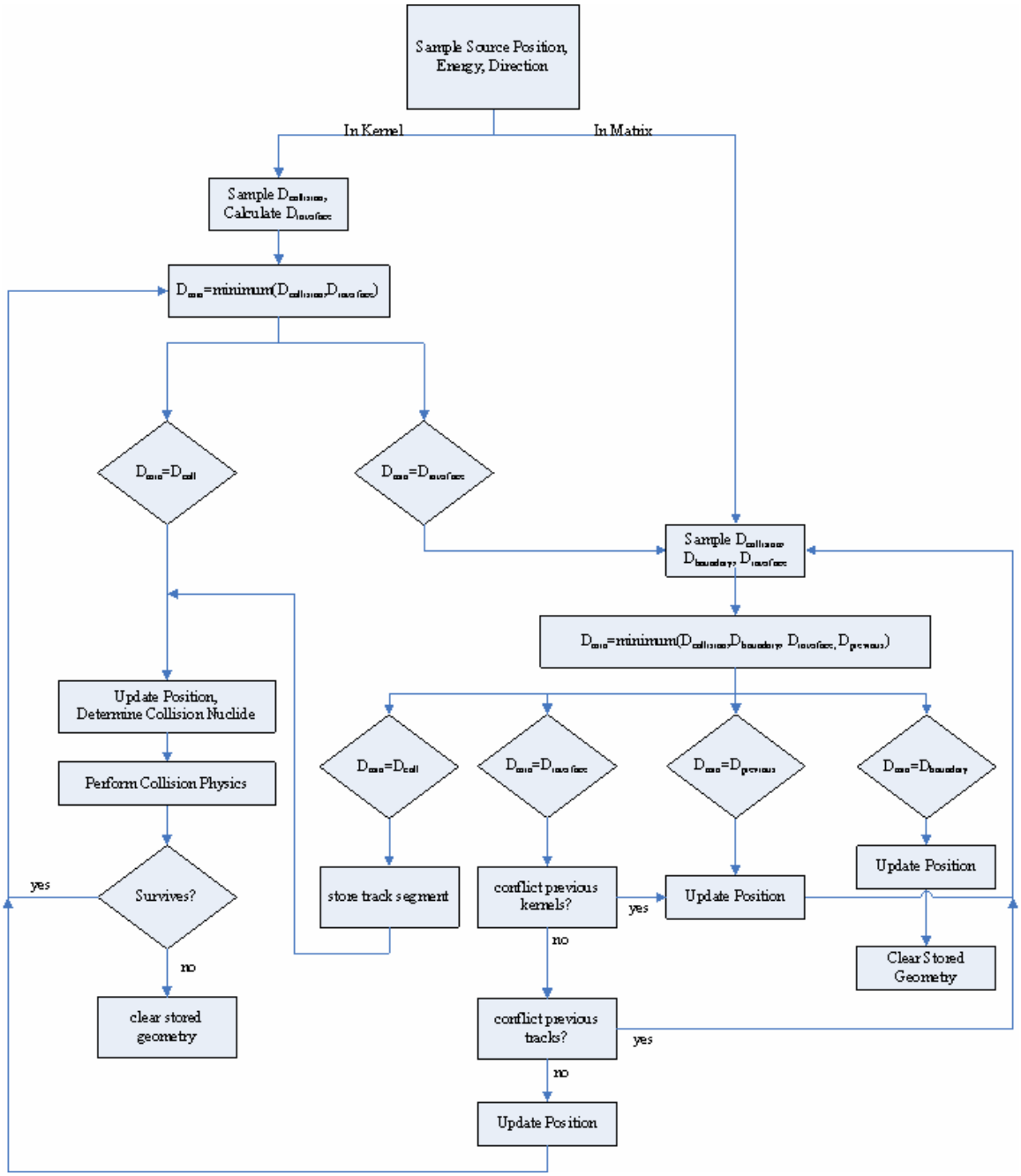
The extended memory chord length sampling technique should be tested for its ability to perform infinite media criticality assessments and its treatment of whole core, double heterogeneous geometries. Calculations that are larger in scope than those examined here will limit the amount of material traversal information for each history that can be efficiently retained in memory. Further optimization of this extended memory chord length sampling method is warranted prior to practical implementation. Future work in the advancement of the resonance switch methodology should test additional fissile materials. This capability would allow the opportunity to perform burnup and depletion calculations of non-fresh fuel. Such an algorithm would address local variations in the isotopic fuel composition. The resonance switch approach should also be validated against other fuel assembly geometries and non-uniform fuel kernels dimensions.

Appendices

APPENDIX A: SAMPLE NJOY INPUT LISTING FOR NATURAL SILICON

```
moder
 20 -21/
 reconr
 -21 -22/
 'pendf for natural silicon'/
 1400 1/
 .05/
 '<pendf>-silicon'/
 0/
 broadr
 -21 -22 -23/
 1400 1/
 .005/
 300/
 0/
 unresr
 -21 -23 -24/
 1400 1 7 1/
 300/
 1.e10 1.e05 1.e03 1.e02 10. 1. .1/
 0/
 groupr
 20 -24 0 -26/
 1400 1 0 3 0 1 7/
 'Error'/
 300/
 1.e10 1.e05 1.e03 1.e02 1.e01 1. .1/
 500/
 100.000000 102.329299 104.712855 107.151931 109.647820.....(to 10 MeV)
 3 1/
 3 2/
 3 102/
 3 251/
 0/
 0/
 moder
 -26 27/
 stop
```

APPENDIX B: LOGIC FLOW FOR EXTENDED MEMORY CHORD LENGTH SAMPLING ALGORITHM



APPENDIX C: MATLAB CODE LISTING FOR EXTENDED MEMORY CHORD LENGTH SAMPLING

```
clear all
format long
rand('twister',sum(100*clock))

%Geometry
R=[2.0];
rad=[0.1];
num_spheres=[600];
cl_avg=[1.3131];

npart=5;   %number of particles to track
nreal=5;   %number of realizations

Ecutoff=0.0001;

%Cross Sections
absorber_xs_factor=[140];
u235total=0.08067*absorber_xs_factor(j);
u235scat=0.03873*absorber_xs_factor(j);
prob_scat235=u235scat/u235total;
u238total=0.35201*absorber_xs_factor(j);
u238scat=0.29845*absorber_xs_factor(j);
prob_scat238=u238scat/u238total;
o16total=0.17904*absorber_xs_factor(j);
o16scat=0.17904*absorber_xs_factor(j);
prob_scato16=o16scat/o16total;
totalxs0=u235total+u238total+o16total;

matrix_xs_factor=[1];
b10total=7.216e-7*matrix_xs_factor(j);
b10scat=5.895e-7*matrix_xs_factor(j);
b11total=4.133e-7*matrix_xs_factor(j);
b11scat=4.133e-7*matrix_xs_factor(j);
ctotal=0.34635*matrix_xs_factor(j);
cscat=0.34635*matrix_xs_factor(j);
totalxs1=b10total+b11total+ctotal;

leakage=zeros(nreal,1);
overlap_count=0;
traversal_err=0;
track_overlap_count=0;

for q=1:nreal

leak=0;
```

```
%Define starting geometry for this realization
[xs_start,ys_start,zs_start] = packsphere_power(num_spheres(j),R,rad(j));

for k=1:npart
    now_index=1;
      xs_now=0;  %reset vectors for storing previous geometry
      ys_now=0;
      zs_now=0;
    track_index=1;
      trackx=0;
      tracky=0;
      trackz=0;
    xpos=0;  %initial particle properties
    ypos=0;
    zpos=0;
    [u,v,w] = isoscat3();
    [E]=wattsource();

% %What material does the particle start in?
is_in_circle=0; %default
    for i=1:num_spheres(j)
        dist_center(i)=sqrt((xpos-xs_start(i))^2+(ypos-ys_start(i))^2+(zpos-zs_start(i))^2);
        if dist_center(i) < rad
           is_in_circle=1;
           m=i;
           xs_now(now_index)=xs_start(m);  %to check for overlap errors later
           ys_now(now_index)=ys_start(m);
           zs_now(now_index)=zs_start(m);
           now_index=now_index+1;
           trackx(track_index)=xpos;
           tracky(track_index)=ypos;
           trackz(track_index)=zpos;
           track_index=track_index+1;
           break
         end
    end



  while E>Ecutoff

     while is_in_circle==1
           [dcoll]=dist_coll(totalxs0);
           [dsphwall]=dist_sph_wall_multi(xpos,ypos,zpos,u,v,w,xs_start,…
           ys_start,zs_start,rad(j),m);
           dist_inside = [dcoll dsphwall];
           dist_ins_min=min(dist_inside);
```

```
        if dcoll==dist_ins_min
              %new location of particle
               xpos=xpos+dcoll*u; ypos=ypos+dcoll*v; zpos=zpos+dcoll*w;
               Atest=rand();
                 if Atest < u235total/totalxs0
                    A=235;
                    c0=prob_scat235;
                 elseif Atest < (u235total+u238total)/totalxs0
                    A=238;
                    c0=prob_scat238;
                 else
                    A=16;
                    c0=prob_scato16;
                 end
                %What type of collision occurs?
                 if rand() < c0 %scatters
                     %Get a new random direction
                     [E,u,v,w]=elastic_scat_Dupree(E,A,u,v,w);
                     trackx(track_index)=xpos;
                     tracky(track_index)=ypos;
                     trackz(track_index)=zpos;
                     track_index=track_index+1;
                     if E < Ecutoff
                        cutoff=cutoff+1;
                     end
                 else  %particle is absorbed
                        E=0;
                        is_in_circle=0;
                 end
         else  %particle leaves birth sphere
              is_in_circle=0;
              xpos=xpos+dsphwall*u*1.0000001; ypos=ypos+dsphwall*v*1.0000001;
             zpos=zpos+dsphwall*w*1.0000001;
        end
     end   %end while in birth circle loop

     if E==0
       break;  %terminate history if captured in starting kernel
     end


%Begin Matrix Tracking Loop
  matrix_temp=0;  %reset following previous capture
  trackx(track_index)=xpos;
  tracky(track_index)=ypos;
  trackz(track_index)=zpos;
  track_index=track_index+1;

  %Now there is possibly another dist check---dist to previous
```

```
dprev=1000; %default=very big value
[dcoll]=dist_coll(totalxs1);
[dwall]=dist_sphere_wall(xpos,ypos,zpos,u,v,w,R);
[dsphere2]=dist_sphere_cl(cl_avg(j));
if xs_now(1)~=0  %if kernels previously located in this history
  [dprev,m]=dist_sphere_multi(xpos,ypos,zpos,u,v,w,xs_now,ys_now,zs_now,rad(j));
end
dist=[dwall dcoll dsphere2 dprev];
min_dist=min(dist);

 %******Distance Decision Statements******
 if dwall==min_dist
    leak=leak+1;  %tally
    E=0;

 elseif dprev==min_dist  %re-entered a previously established sphere
       is_in_circle=1;
       xs=xs_now(m);
       ys=ys_now(m);
       zs=zs_now(m);
       xpos=xpos+dprev*u*1.0000001; ypos=ypos+dprev*v*1.0000001;
      zpos=zpos+dprev*w*1.0000001;

       %Check that the particle is indeed inside the sphere
       sphere_dist=((xpos-xs)^2+(ypos-ys)^2+(zpos-zs)^2)^0.5;
       if sphere_dist >= rad(j)
         us=(xs-xpos)/sphere_dist; %direction from sphere center to current position
         vs=(ys-ypos)/sphere_dist;
         ws=(zs-zpos)/sphere_dist;
         xpos=xs+0.9999*us*rad(j);
         ypos=ys+0.9999*vs*rad(j);
         zpos=zs+0.9999*ws*rad(j);
       end

       while is_in_circle==1
          [dcoll]=dist_coll(totalxs0);
          [dsphwall]=dist_sph_wall_cl_explicit(xpos,ypos,zpos,u,v,w,xs,ys,zs,rad(j));
          dist_inside = [dcoll dsphwall];
          dist_ins_min=min(dist_inside);

          if dcoll==dist_ins_min
           xpos=xpos+dcoll*u; ypos=ypos+dcoll*v; zpos=zpos+dcoll*w;
           %Determine which nucleus the particle collided with
            Atest=rand();
              if Atest < u235total/totalxs0
                 A=235;
                 c0=prob_scat235;
              elseif Atest < (u235total+u238total)/totalxs0
                 A=238;
```

```
            c0=prob_scat238;
          else
            A=16;
            c0=prob_scato16;
          end

          %What type of collision occurs?
          if rand() < c0  %scatters
                %Get a new random direction
                [E,u,v,w]=elastic_scat_Dupree(E,A,u,v,w);
                trackx(track_index)=xpos;
                tracky(track_index)=ypos;
                trackz(track_index)=zpos;
                track_index=track_index+1;
                if E < Ecutoff
                  cutoff=cutoff+1;
                end
             else  %particle is absorbed in a fission or
                  %capture
                  E=0;  %history will be terminate regardless of capture or fission
                  is_in_circle=0; %terminate loop
                  capt=capt+1;
                  prev_cap=prev_cap+1;
          end

        else  %particle leaves the interstitial sphere
            is_in_circle=0;
            xpos=xpos+dsphwall*u*1.0000001;ypos=ypos+dsphwall*…
             v*1.0000001;  zpos=zpos+dsphwall*w*1.0000001;
       end
    end   %end while in circle loop;

elseif dcoll==min_dist
    xpos=xpos+dcoll*u; ypos=ypos+dcoll*v; zpos=zpos+dcoll*w;
    Atest=rand();
    if Atest < ctotal/totalxs1
      A=12;
      c1=cscat/ctotal;
    elseif Atest < (ctotal+b10total)/totalxs1
      A=10;
      c1=b10scat/b10total;
    else
      A=11;
      c1=b11scat/b11total;
    end

    if rand()<c1 %scatters
      [E,u,v,w]=elastic_scat_Dupree(E,A,u,v,w);
      trackx(track_index)=xpos;
```

```
        tracky(track_index)=ypos;
        trackz(track_index)=zpos;
        track_index=track_index+1;
      else  %particle is absorbed--no chance to fission in graphite
         E=0;
         capt=capt+1;
      end

else  %particle enters a new CLS sphere
      is_in_circle=1;
      overlap=0; %defaults for overlap check
      track_overlap=0;
      %Do not assign position permanently until we know sphere does not overlap
      xpostemp=xpos+dsphere2*u*1.0000001; ypostemp=ypos+dsphere2*…
      v*1.0000001; zpostemp=zpos+dsphere2*w*1.0000001;

        %Assign Coordinates for Center of stochastic sphere
            xinterface=xpostemp; yinterface=ypostemp; zinterface=zpostemp;
            [cl_sphere]=sphere_cl(rad(j));
            rads=sqrt(rad(j)^2-(cl_sphere/2)^2);
            xchordmid=xinterface+(cl_sphere/2)*u;
            ychordmid=yinterface+(cl_sphere/2)*v;
            zchordmid=zinterface+(cl_sphere/2)*w;

            %Transform sphere coordinates
            theta_cl=2*pi*rand();
            uprimecl=cos(theta_cl);
            vprimecl=sin(theta_cl);
            wprimecl=0;
            dsq=sqrt(u^2+v^2);
            xs=xchordmid+rads*(v/dsq*uprimecl+w*u/dsq*vprimecl);
            ys=ychordmid+rads*(-u/dsq*uprimecl+w*v/dsq*vprimecl);
            zs=zchordmid+rads*(-dsq*vprimecl);

            %Check to see if newest sphere
            %overlaps any previous
              if xs_now ~= 0
                for i=1:length(xs_now)
                   if (xs-xs_now(i))^2+(ys-ys_now(i))^2+(zs-zs_now(i))^2 …
                       < (2*rad(j))^2
                      overlap_count=overlap_count+1;
                      overlap=1;
                   end
                end

              end

              if length(trackx)>=2
              [track_overlap]=tracks(trackx,tracky,trackz,xs,ys,zs,rad(j));
```

```
          end
          %Does the sphere overlap the pebble
          %boundary
          if (xs^2+ys^2+zs^2) > (R-rad(j))^2
             is_in_circle=0;
             xpos=xpostemp; ypos=ypostemp; zpos=zpostemp;
          elseif overlap==1
             %sphere conflicts previous placed
             is_in_circle=0;
          elseif track_overlap==1
             %sphere conflicts previous track segment
             is_in_circle=0;
          else %sphere location is ok
             xpos=xpostemp; ypos=ypostemp; zpos=zpostemp;
             xs_now(now_index)=xs;  %to check for overlap errors later
             ys_now(now_index)=ys;
             zs_now(now_index)=zs;
             now_index=now_index+1;
          end

while is_in_circle==1
   [dcoll]=dist_coll(totalxs0);
   [dsphwall]=dist_sph_wall_cl_explicit(xpos,ypos,zpos,u,v,w,…
     xs,ys,zs,rad(j));
   dist_inside = [dcoll dsphwall];
   dist_ins_min=min(dist_inside);

if dcoll==dist_ins_min
   xpos=xpos+dcoll*u; ypos=ypos+dcoll*v; zpos=zpos+dcoll*w;
   Atest=rand();
   if Atest < u235total/totalxs0
     A=235;
     c0=prob_scat235;
   elseif Atest < (u235total+u238total)/totalxs0
     A=238;
     c0=prob_scat238;
   else
     A=16;
     c0=prob_scato16;
     scattertest=1;
   end

    %What type of collision occurs?
    if rand() < c0  %scatters
          %Get a new random direction
          [E,u,v,w]=elastic_scat_Dupree(E,A,u,v,w);
          trackx(track_index)=xpos;
          tracky(track_index)=ypos;
          trackz(track_index)=zpos;
```

```
                    track_index=track_index+1;
                    if E < Ecutoff
                      cutoff=cutoff+1;
                    end
                  else  %particle is absorbed
                      E=0;  %history will be terminate regardless of capture or fission
                      is_in_circle=0; %terminate loop
                end

              else  %particle leaves the interstitial sphere
                   is_in_circle=0;
                   xpos=xpos+dsphwall*u*1.0000001; pos=ypos+dsphwall*v*…
                    1.0000001; zpos=zpos+dsphwall*w*1.0000001;
             end
          end   %end while in circle loop;
    end %end main matrix distance check if statement
   end %end of that history
end %end of total particles to be tracked for that batch  (npart)

leakage(q)=leak;
end %end all realizations

%Compile Results
leakage_fraction=leakage/(npart);
leakage_fraction=mean(leakage_fraction)
```

APPENDIX D: MATLAB CODE LISTING FOR PEAK BROADENED OPTICAL THICKNESS THRESHOLD RESONANCE SWITCH ALGORITHM

```
clear all
format long
rand('twister',sum(100*clock))

%Geometry
R=2.5;   %radius of bounding sphere
num_spheres=9394;
rad=0.0251;
Rshell=3.0;  %radius of matrix shell+
Rmod=3.7959;
V0=(4/3)*num_spheres*pi*rad^3;
V1=(4/3)*pi*R^3-V0;
VF_kernel=V0/(V1+V0);

%Define Geometry
[xs]=xs_save();
[ys]=ys_save();
[zs]=zs_save();

%Set parameters for batch traces
npart=1000   %number of particles to track per batch
ntransient=5;
nbatches=20; %additional batches
opt_threshold=[1.0]

collision_k=zeros(1,ntransient+nbatches);

%Call cross section group structure
[egroup]=egroupstructure();
[inelastic_group]=inelasticgroupstructure();
[u235_inelastic_matrix]=u235inelasticmatrix();
[u238_inelastic_matrix]=u238inelasticmatrix();
Ecut=egroup(2); %energy cutoff pt.
bol=8.617342e-5;
temp=300;

%Define Number Densities
%Kernel
Nu235=4.11729e-3;
Nu238=2.01906e-2;
No16=4.86157e-2;
%Buffer
Nb10buffer=1.58108e-8;
```

```
Nb11buffer=6.40418e-8;
Ncbuffer=5.51511e-2;
%Inner PyC
Nb10pyc1=2.73095e-8;
Nb11pyc1=1.10617e-7;
Ncpyc1=9.52609e-2;
%SiC
Nb10sic=1.38e-8;
Nb11sic=5.5808e-8;
Ncsic=4.80603e-2;
Nsisic=4.80603e-2;
%Outer Pyc
Nb10pyc2=2.73095e-8;
Nb11pyc2=1.10617e-7;
Ncpyc2=9.52609e-2;
%Matrix
Nb10matrix=2.47855e-8;
Nb11matrix=1.00394e-7;
Ncmatrix=8.64563e-2;
%Shell
Nb10shell=2.47855e-8;
Nb11shell=1.00394e-7;
Ncshell=8.64563e-2;
%Moderator/Air
Nh1mod=2.97225e-7;
Nh2mod=4.45905e-11;
Nb10mod=1.17151e-8;
Nb11mod=4.7452e-8;
Ncmod=4.08644e-2;
Nn14mod=2.0426e-5;
Nn15mod=7.5035e-8;
No16mod=5.1253e-6;
%Homogenized Matrix Volumes
Vbuffer=1.02056e-4;
Vpyc1=6.59352e-5;
Vsic=7.10387e-5;
Vpyc2=9.63663e-5;
Vmatrix=6.56779e-3;
Vtotal=Vbuffer+Vpyc1+Vsic+Vpyc2+Vmatrix;
Nb10hom=(Nb10buffer*Vbuffer+Nb10pyc1*Vpyc1+Nb10sic*Vsic+Nb10pyc2*Vpyc2+Nb1
0matrix*Vmatrix)/Vtotal;
Nb11hom=(Nb11buffer*Vbuffer+Nb11pyc1*Vpyc1+Nb11sic*Vsic+Nb11pyc2*Vpyc2+Nb1
1matrix*Vmatrix)/Vtotal;
Nchom=(Ncbuffer*Vbuffer+Ncpyc1*Vpyc1+Ncsic*Vsic+Ncpyc2*Vpyc2+Ncmatrix*Vmatrix
)/Vtotal;
Nsihom=(Nsisic*Vsic)/Vtotal;

%Determine optical thickness of interstitital
[opt_thickness] = optical_thickness(rad,Nu235,Nu238,No16);
```

```
G=length(opt_thickness);
opt_step=zeros(1,G*2);
E_step=zeros(1,G*2);
opt_step(1)=opt_thickness(1);
E_step(1)=egroup(1);

for(j=1:G-1)
    opt_step(2*j)=opt_thickness(j);
    opt_step(2*j+1)=opt_thickness(j+1);
end
opt_step(G*2)=opt_thickness(G);

for(j=1:G-1)
   E_step(2*j)=egroup(j+1);
   E_step(2*j+1)=egroup(j+1);
end
E_step(G*2)=10.0e6;

opt_bins=length(egroup);
bin_value=zeros(1,length(egroup));
for h=35:opt_bins  %exclude thermal energies
   if opt_thickness(h) > opt_threshold(q)
      bin_value(h)=opt_thickness(h);
   end
end
bin_value_orig=bin_value;

E_lower=egroup;
E_upper=zeros(length(egroup),1);
for k=1:length(egroup)-1
    E_upper(k)=egroup(k+1);
end
E_upper(length(egroup))=10e6;

bin_value_new=zeros(1,length(egroup));

%Mark the groups immediately adjacent to the res peak as dark groups
for k=2:length(bin_value)-1
    if bin_value_orig(k)~=0
       bin_value(k-1)=opt_thickness(k-1);
       bin_value(k+1)=opt_thickness(k+1);
       bin_value_new(k-1)=opt_thickness(k-1);
       bin_value_new(k+1)=opt_thickness(k+1);
    end
end


%Start Resonance Switch Calculation
```

```
%Initialize Fission Banks
   xbank=zeros(npart,2); ybank=zeros(npart,2); zbank=zeros(npart,2);
   xbanknorm=zeros(npart,2); ybanknorm=zeros(npart,2); zbanknorm=zeros(npart,2);

resstart=clock;
for j=1:ntransient+nbatches
   xbank=xbanknorm; ybank=ybanknorm; zbank=zbanknorm;

for k=1:npart
   if j==1
     %Initial Source Distribution
      is_in_res=0; %default assumptions
      [E]=wattsource();
      [xpos,ypos,zpos,m,u,v,w] = source1_power_fission(xs,ys,zs,rad);
   else
     %Source Distribution sample from previous generation
     %fission sites
     [E]=wattsource();
       %*****************************************************************
        %Are we in the resonance range?
           for i=1:length(egroup)
             %Find the index of the bin lower bound
              if egroup(i) > E
                 bin_index=i-1;
                 break
              end
           end
           %Is this bin one that is flagged as a resonance?
           if bin_value(bin_index)~=0
             is_in_res=1;
           else
              is_in_res=0;
           end
       %*****************************************************************
     if is_in_res==0
        [xpos,ypos,zpos,u,v,w] = source3am(xbank,ybank,zbank,k,R);
     else
        [xpos,ypos,zpos,m,u,v,w] = source3(xbank,ybank,zbank,k,xs,ys,zs,R,rad);
     end
   end

  %Where does the history start
  is_in_matrix=1; is_in_shell=0; is_in_mod=0;
  etabank(k)=0;

while E>Ecut
```

```
if is_in_res==1 && is_in_matrix==1 %if in matrix or an actual kernel

    while is_in_res==1 && is_in_matrix==1
        %Begin Matrix Tracking Loop
        %Begin Woodcock Tracking
        [homb10total]=b10totalxs(E,egroup)*Nb10hom;
        [homb11total]=b11totalxs(E,egroup)*Nb11hom;
        [homctotal]=ctotalxs(E,egroup)*Nchom;
        [homsitotal]=sitotalxs(E,egroup)*Nsihom;
        totalxs1=homb10total+homb11total+homctotal+homsitotal;
        [homb10scat]=b10scatxs(E,egroup)*Nb10hom;
        [homb11scat]=b11scatxs(E,egroup)*Nb11hom;
        [homcscat]=cscatxs(E,egroup)*Nchom;
        [homsiscat]=siscatxs(E,egroup)*Nsihom;
        scatterxs1=homb10scat+homb11scat+homcscat+homsiscat;

        [u235total]=u235totalxs(E,egroup)*Nu235;
        [u238total]=u238totalxs(E,egroup)*Nu238;
        [o16total]=o16totalxs(E,egroup)*No16;
        totalxs0=u235total+u238total+o16total;
        [u235scat]=u235scatxs(E,egroup)*Nu235;
        [u238scat]=u238scatxs(E,egroup)*Nu238;
        [o16scat]=o16scatxs(E,egroup)*No16;
        scatterxs0=u235scat+u238scat+o16scat;
        [u235inelastic]=u235inelasticxs(E,egroup)*Nu235;
        [u238inelastic]=u238inelasticxs(E,egroup)*Nu235;
        u235scat_total=u235scat+u235inelastic;
        u238scat_total=u238scat+u238inelastic;
        [u235fission]=u235fissionxs(E,egroup)*Nu235;
        [u238fission]=u238fissionxs(E,egroup)*Nu238;
        fissionxs0=u235fission+u238fission;
        [u235absorption]=u235absorptionxs(E,egroup)*Nu235;
        [u238absorption]=u238absorptionxs(E,egroup)*Nu238;
        [o16absorption]=o16absorptionxs(E,egroup)*No16;
        absorptionxs0=u235absorption+u238absorption+o16absorption;
        [nu235]=nu235xs(E,egroup);
        [nu238]=nu238xs(E,egroup);
        prob_fiss235=u235fission/u235total;
        prob_fiss238=u238fission/u238total;
        prob_abs235=u235absorption/u235total;
        prob_abs238=u238absorption/u238total;
        prob_abso16=o16absorption/o16total;
        prob_scat235=1-prob_abs235;
        prob_scat238=1-prob_abs238;
        prob_scato16=1-prob_abso16;

        maxvec=[totalxs0 totalxs1];
        maxed=max(maxvec);
```

```
    [dcoll]=dist_coll(maxed);
    [dwall]=dist_sphere_wall(xpos,ypos,zpos,u,v,w,R); %distance to wall
    dist=[dwall dcoll];
    min_dist=min(dist);

%******Distance Decision Statements******
        if dwall==min_dist
                is_in_shell=1; is_in_matrix=0;
                xpos=xpos+dwall*u*1.00001; ypos=ypos+dwall*v*1.00001;…
                zpos=zpos+dwall*w*1.00001;

        else % dcoll==min_dist
                xpos=xpos+dcoll*u; ypos=ypos+dcoll*v; zpos=zpos+dcoll*w;

                %What material is the particle in?
                totalxs=totalxs1;  %by default assumes its in matrix
                for i=1:num_spheres
                   dist_center(i)=sqrt((xpos-xs(i))^2+(ypos-ys(i))^2+(zpos-zs(i))^2);
                   if dist_center(i) < rad
                      totalxs=totalxs0;
                      break
                   end
                end

                %Is the collision real?
                zeta2=rand();
                if zeta2 < totalxs/maxed
                  %Collision is real
                  if totalxs==totalxs1
                  %Real collision occurred in matrix
                     %Determine which nucleus the particle collided with
                      Atest=rand();
                      if Atest < homctotal/totalxs1
                         A=12;
                         c1=homcscat/homctotal;
                      elseif Atest < (homctotal+homb10total)/totalxs1
                         A=10;
                         c1=homb10scat/homb10total;
                      elseif Atest < (homctotal+homb10total+homsitotal)/totalxs1
                         A=28;
                         c1=homsiscat/homsitotal;
                      else
                         A=11;
                         c1=homb11scat/homb11total;
                      end

                      if rand()<c1 %scatters
                           if E >16*bol*temp
                             [E,u,v,w]=elastic_scat_Dupree(E,A,u,v,w);
```

```
                    else
                      [u,v,w,E]=therm(u,v,w,E,temp,A);
                    end
                    %*************************************************************
                     if E>Ecut
                     %Are we in the resonance range?
                        for i=1:length(egroup)
                          %Find the index of the bin lower bound
                           if egroup(i) > E
                              bin_index=i-1;
                              break
                           end
                        end

                        %Is this bin one that is flagged as a resonance?
                        if bin_value(bin_index)~=0
                          is_in_res=1;
                        else
                          is_in_res=0;
                        end
                     end
%*************************************************************

                  else  %particle is absorbed--no chance to fission in graphite
                     E=0;
                  end

               else  %totalxs=totalxs0
                  %Real collision occurred in a kernel
                   %Determine which nucleus the particle collided with
                   Atest=rand();
                    if Atest < u235total/totalxs0
                      A=235;
                      c0=prob_scat235;
                      scattertest=u235scat/u235scat_total;
                      inelastic_matrix=u235_inelastic_matrix;
                      collision_k(j)=collision_k(j)+(nu235*u235fission)/u235total;
                    elseif Atest < (u235total+u238total)/totalxs0
                      A=238;
                      c0=prob_scat238;
                      scattertest=u238scat/u238scat_total;
                      inelastic_matrix=u238_inelastic_matrix;
                      collision_k(j)=collision_k(j)+(nu238*u238fission)/u238total;
                    else
                      A=16;
                      c0=prob_scato16;
                      scattertest=1;
                    end
```

```
                %What type of collision occurs?
               if rand() < c0 %scatters
                    if rand() < scattertest %elastic
                        if E >16*bol*temp
                          [E,u,v,w]=elastic_scat_Dupree(E,A,u,v,w);
                        else
                          [u,v,w,E]=therm(u,v,w,E,temp,A);
                        end
                    else %inelastic
                       [u,v,w,E]=inelastic(E,inelastic_group,inelastic_matrix);
                    end

%*************************************************************
                     if E>Ecut
                     %Are we in the resonance range?
                        for i=1:length(egroup)
                          %Find the index of the bin lower bound
                           if egroup(i) > E
                              bin_index=i-1;
                              break
                           end
                        end

                        %Is this bin one that is flagged as a resonance?
                        if bin_value(bin_index)~=0
                          is_in_res=1;
                        else
                          is_in_res=0;
                        end
                     end
%*************************************************************

                 else  %particle is absorbed in a fission or
                     %capture
                        if A==235
                          [nu]=nu235xs(E,egroup);
                            etabank(k)=nu*(prob_fiss235/prob_abs235);
                            xbank(k,2)=xpos; ybank(k,2)=ypos; zbank(k,2)=zpos;
                        elseif A==238
                          [nu]=nu238xs(E,egroup);
                            etabank(k)=nu*(prob_fiss238/prob_abs238);
                            xbank(k,2)=xpos; ybank(k,2)=ypos; zbank(k,2)=zpos;
                        end
                    E=0;  %history will be terminate regardless of capture or fission
                    is_in_matrix=0;
               end
            end

          else
```

```
            %collision is not real
          end

      end %end main distance check if statement
   end %end while in matrix and resonance


elseif is_in_shell==1
          while is_in_shell==1
            [b10total]=b10totalxs(E,egroup)*Nb10shell;
            [b11total]=b11totalxs(E,egroup)*Nb11shell;
            [ctotal]=ctotalxs(E,egroup)*Ncshell;
            totalxs1=b10total+b11total+ctotal;
            [b10scat]=b10scatxs(E,egroup)*Nb10shell;
            [b11scat]=b11scatxs(E,egroup)*Nb11shell;
            [cscat]=cscatxs(E,egroup)*Ncshell;
            scatterxs1=b10scat+b11scat+cscat;

            [dshellwall]=dist_shell(xpos,ypos,zpos,u,v,w,Rshell);
            [dcoll]=dist_coll(totalxs1);
            [dmatrix]=dist_matrix(xpos,ypos,zpos,u,v,w,R);

            dist=[dshellwall dcoll dmatrix];
            min_dist=min(dist);

            if dcoll==min_dist
              Atest=rand();
              if Atest < ctotal/totalxs1
                    A=12;
                    c1=cscat/ctotal;
              elseif Atest < (ctotal+b10total)/totalxs1
                    A=10;
                    c1=b10scat/b10total;
              else
                    A=11;
                    c1=b11scat/b11total;
              end

              if rand()<c1 %scatters
                   if E >16*bol*temp
                     [E,u,v,w]=elastic_scat_Dupree(E,A,u,v,w);
                   else
                     [u,v,w,E]=therm(u,v,w,E,temp,A);
                   end
              else  %particle is absorbed--no chance to fission in graphite
                 E=0;
                 is_in_shell=0;
              end
```

```
        elseif dshellwall==min_dist
              is_in_shell=0; is_in_mod=1;
              xpos=xpos+dshellwall*u*1.000001; ypos=ypos+…
             dshellwall*v*1.000001; zpos=zpos+dshellwall*w*1.000001;

        else %particle leaves the shell region and leaks back into matrix
             is_in_shell=0; is_in_matrix=1;
             xpos=xpos+dmatrix*u*1.000001; ypos=ypos+.dmatrix*v*…
             1.000001; zpos=zpos+dmatrix*w*1.000001;
        end %end distance comparison
    end %end while in shell


elseif is_in_mod==1

    while is_in_mod==1
        [dmodwall]=dist_shell(xpos,ypos,zpos,u,v,w,Rmod);
        [h1modtotal]=h1totalxs(E,egroup)*Nh1mod;
        [h2modtotal]=h2totalxs(E,egroup)*Nh2mod;
        [b10modtotal]=b10totalxs(E,egroup)*Nb10mod;
        [b11modtotal]=b11totalxs(E,egroup)*Nb11mod;
        [cmodtotal]=ctotalxs(E,egroup)*Ncmod;
        [n14modtotal]=n14totalxs(E,egroup)*Nn14mod;
        [n15modtotal]=n15totalxs(E,egroup)*Nn15mod;
        [h1modscat]=h1scatxs(E,egroup)*Nh1mod;
        [h2modscat]=h2scatxs(E,egroup)*Nh2mod;
        [b10modscat]=b10scatxs(E,egroup)*Nb10mod;
        [b11modscat]=b11scatxs(E,egroup)*Nb11mod;
        [cmodscat]=cscatxs(E,egroup)*Ncmod;
        [n14modscat]=n14scatxs(E,egroup)*Nn14mod;
        [n15modscat]=n15scatxs(E,egroup)*Nn15mod;
        totalxs1=h1modtotal+h2modtotal+b10modtotal+b11modtotal+…
        cmodtotal+n14modtotal+n15modtotal;

        [dcoll]=dist_coll(totalxs1);
        [dmatrix]=dist_matrix(xpos,ypos,zpos,u,v,w,Rshell);

        dist=[dmodwall dcoll dmatrix];
        min_dist=min(dist);

        if dcoll==min_dist
          xpos=xpos+dcoll*u; ypos=ypos+dcoll*v; zpos=zpos+dcoll*w;

          %Determine which nucleus the particle collided with
          Atest=rand();
          if Atest < h1modtotal/totalxs1
            A=1;
            c1=h1modscat/h1modtotal;
          elseif Atest < (h1modtotal+h2modtotal)/totalxs1
```

```
        A=2;
        c1=h2modscat/h2modtotal;
      elseif Atest < (h1modtotal+h2modtotal+b10modtotal)/totalxs1
        A=10;
        c1=b10modscat/b10modtotal;
      elseif Atest < (h1modtotal+h2modtotal+b10modtotal+…
                      b11modtotal)/totalxs1
        A=11;
        c1=b11modscat/b11modtotal;
      elseif Atest < (h1modtotal+h2modtotal+b10modtotal+b11modtotal+…
                      cmodtotal)/totalxs1
        A=12;
        c1=cmodscat/cmodtotal;
      elseif Atest < (h1modtotal+h2modtotal+b10modtotal+b11modtotal+…
                      n14modtotal)/totalxs1
        A=14;
        c1=n14modscat/n14modtotal;
      elseif Atest < (h1modtotal+h2modtotal+b10modtotal+b11modtotal+…
                     n14modtotal+n15modtotal)/totalxs1
        A=15;
        c1=n15modscat/n15modtotal;
      else
        A=16;
        c1=o16scat/o16total;
      end

      if rand()<c1 %scatters
        if E > 16*bol*temp
          [E,u,v,w]=elastic_scat_Dupree(E,A,u,v,w);
        else
          [u,v,w,E]=therm(u,v,w,E,temp,A);
        end

      else  %particle is absorbed--no chance to fission in moderator
          E=0;
          is_in_mod=0;
      end

    elseif dmodwall==min_dist
          xpos=xpos+dmodwall*u*0.9999999; ypos=ypos+…
         dmodwall*v*0.9999999; zpos=zpos+dmodwall*w*0.9999999;
          [u,v,w]=whitereflection3(xpos,ypos,zpos,u,v,w);

    else %particle leaves the moderator region and leaks back into shell
          is_in_mod=0; is_in_shell=1;
          xpos=xpos+dmatrix*u*1.000001; ypos=ypos+dmatrix*v*1.000001;…
           zpos=zpos+dmatrix*w*1.000001;
    end
end
```

```
%%%Atomic Mix

else %is_in_res==0 && is_in_matrix==1 %start AM tracking

      %Check that the particle is indeed inside the matrix
            origin_dist=((xpos)^2+(ypos)^2+(zpos)^2)^0.5;
            if origin_dist >= R %still outside the shell
               us=(xpos)/origin_dist; %direction from sphere center to current position
               vs=(ypos)/origin_dist;
               ws=(zpos)/origin_dist;
               xpos=0.9999*us*R;
               ypos=0.9999*vs*R;
               zpos=0.9999*ws*R;
             end

      while is_in_matrix==1 && is_in_res==0
         %Kernel Cross Sections
         [u235total]=u235totalxs(E,egroup)*Nu235;
         [u238total]=u238totalxs(E,egroup)*Nu238;
         [o16total]=o16totalxs(E,egroup)*No16;
         totalxs0=u235total+u238total+o16total;
         [u235scat]=u235scatxs(E,egroup)*Nu235;
         [u238scat]=u238scatxs(E,egroup)*Nu238;
         [o16scat]=o16scatxs(E,egroup)*No16;
         scatterxs0=u235scat+u238scat+o16scat
         [u235inelastic]=u235inelasticxs(E,egroup)*Nu235;
         [u238inelastic]=u238inelasticxs(E,egroup)*Nu235;
         u235scat_total=u235scat+u235inelastic;
         u238scat_total=u238scat+u238inelastic;
         [u235fission]=u235fissionxs(E,egroup)*Nu235;
         [u238fission]=u238fissionxs(E,egroup)*Nu238;
         fissionxs0=u235fission+u238fission;
         [u235absorption]=u235absorptionxs(E,egroup)*Nu235;
         [u238absorption]=u238absorptionxs(E,egroup)*Nu238;
         [o16absorption]=o16absorptionxs(E,egroup)*No16;
         absorptionxs0=u235absorption+u238absorption+o16absorption;
         [nu235]=nu235xs(E,egroup);
         [nu238]=nu238xs(E,egroup);
         nu_sigf=nu235*u235fission+nu238*u238fission;

         %Matrix Cross Sections
         [homb10total]=b10totalxs(E,egroup)*Nb10hom;
         [homb11total]=b11totalxs(E,egroup)*Nb11hom;
         [homctotal]=ctotalxs(E,egroup)*Nchom;
         [homsitotal]=sitotalxs(E,egroup)*Nsihom;
         totalxs1=homb10total+homb11total+homctotal+homsitotal;
         [homb10scat]=b10scatxs(E,egroup)*Nb10hom;
         [homb11scat]=b11scatxs(E,egroup)*Nb11hom;
```

```
    [homcscat]=cscatxs(E,egroup)*Nchom;
    [homsiscat]=siscatxs(E,egroup)*Nsihom;
    scatterxs1=homb10scat+homb11scat+homcscat+homsiscat;

 %Atomic Mix Homogenized Cross Sections
    M_xsec_total=(V0*totalxs0+V1*totalxs1)/(V1+V0);
    prob235=u235total*V0/((V1+V0)*M_xsec_total);
    prob238=u238total*V0/((V1+V0)*M_xsec_total);
    probo16=o16total*V0/((V1+V0)*M_xsec_total);
    probc=homctotal*V1/((V1+V0)*M_xsec_total);
    probb10=homb10total*V1/((V1+V0)*M_xsec_total);
    probb11=homb11total*V1/((V1+V0)*M_xsec_total);
    probsi=homsitotal*V1/((V1+V0)*M_xsec_total);

[dcoll]=dist_coll(M_xsec_total);
[dwall]=dist_sphere_wall(xpos,ypos,zpos,u,v,w,R); %distance to wall
dist=[dwall dcoll];
min_dist=min(dist);

if dwall==min_dist
   is_in_shell=1; is_in_matrix=0;
   xpos=xpos+dwall*u*1.000001; ypos=ypos+dwall*v*1.000001;…
   zpos=zpos+dwall*w*1.000001;

else   %dcoll==min_dist
       xpos=xpos+dcoll*u; ypos=ypos+dcoll*v; zpos=zpos+dcoll*w;
       Atest=rand();
       if Atest < probc
            A=12;
            c1=homcscat/homctotal;
            scattertest=1;  %probability of scatter being elastic
          elseif Atest < probc+probb10
            A=10;
            c1=homb10scat/homb10total;
            scattertest=1;
          elseif Atest < probc+probb10+prob235
            A=235;
            c1=u235scat_total/u235total;
            scattertest=u235scat/u235scat_total;
            inelastic_matrix=u235_inelastic_matrix;
            collision_k(j)=collision_k(j)+(nu235*u235fission)/u235total;
          elseif Atest < probc+probb10+prob235+prob238
            A=238;
            c1=u238scat_total/u238total;
            scattertest=u238scat/u238scat_total;
            inelastic_matrix=u238_inelastic_matrix;
            collision_k(j)=collision_k(j)+(nu238*u238fission)/u238total;
          elseif Atest < probc+probb10+prob235+prob238+probo16
            A=16;
```

```
            c1=o16scat/o16total;
            scattertest=1;
        elseif Atest < probc+probb10+prob235+prob238+probo16+probsi
            A=28;
            ca=homsiscat/homsitotal;
            scattertest=1;
        else
            A=11;
            c1=homb11scat/homb11total;
            scattertest=1;
    end

    if rand()<c1 %scatters
        if rand() < scattertest %elastic
            if E >16*bol*temp
                [E,u,v,w]=elastic_scat_Dupree(E,A,u,v,w);
            else
                [u,v,w,E]=therm(u,v,w,E,temp,A);
            end
        else %inelastic
            [u,v,w,E]=inelastic(E,inelastic_group,inelastic_matrix);
        end

            %*****************************************************************
            if E>Ecut
            %Are we in the resonance range?
                for i=1:length(egroup)
                  %Find the index of the bin lower bound
                   if egroup(i) > E
                      bin_index=i-1;
                      break
                   end
                end

                %Is this bin one that is flagged as a resonance?
                if bin_value(bin_index)~=0
                  is_in_res=1;
                else
                  is_in_res=0;
                end
              end
%*****************************************************************

    else  %particle is absorbed in a fission or capture
        if A==235
            [nu]=nu235xs(E,egroup);
             etabank(k)=nu*(u235fission/u235absorption);
             xbank(k,2)=xpos; ybank(k,2)=ypos; zbank(k,2)=zpos;
           elseif A==238
```

```
                    [nu]=nu238xs(E,egroup);
                    etabank(k)=nu*(u238fission/u238absorption);
                    xbank(k,2)=xpos; ybank(k,2)=ypos; zbank(k,2)=zpos;
                  end
                  E=0;  %history wfill be terminate regardless of capture or fission
                  is_in_matrix=0;
                end %end if scatters

            end %end of check distance loop
        end %end of while in matrix and NOT resonance
    end %end of check resonance and spatial region
  end %end of that particle history--energy cutoff
end %end of all particles for A.M. batch

%Normalize for next batch
[xbanknorm,ybanknorm,zbanknorm,keffres]=source2_Dupree_mod(xbank,ybank,zbank,
npart,etabank);
%Save Results to Plot
kplot(j)=keffres;
cumulativeplot(j)=mean(kplot);
end %end total number of batches (ntransient+nbatches)
resstop=clock;
resswitchtime=etime(resstop,resstart)
timeper=resswitchtime/(npart*(ntransient+nbatches))
res_k=cumulativeplot(j)

collision_k=collision_k';
collision_k=collision_k/npart
collision_k=mean(collision_k)

overall_k=(res_k+collision_k)/2
```

References

Adams, M., Larsen, E, and Pomraning, G. "Benchmark Results for particle transport in a binary Markov statistical medium, Journal of Quantitative Spectroscopy and Radiative Transfer, Vol. 70, 115-128 (1989).

Bende, E. Plutonium Burning in a Pebble-Bed Type High Temperature Nuclear Reactor, Ph.D. Thesis, Delft University of Technology (1999).

Brown, F. " Fundamentals of Monte Carlo Particle Transport," Los Alamos National Laboratory, LA-UR-05-4983 (2005).

Brown, F. and Martin, W., "Stochastic geometry capability in MCNP5 for the analysis of particle fuel," Annals of Nuclear Energy, Vol. 31 (2004).

Brown, F. *et al*. "A General Monte-Carlo N-Particle Transport Code Version 5", Los Alamos National Laboratory (2003).

Brown, F. *et al*.. "Continuously Varying Materials Properties and Calculations for Monte Carlo Calculations," PHYSOR 2004, Chicago, Illinois (2004b).

Chadwick, M. *et al*. "ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology, Nuclear Data Sheets, Volume 107, Issue 12, 2931-3060 (2006).

Davis, I., Palmer, T. and Larsen, E. "A Comparison of Binary Stochastic Media Transport Models in "Solid-Void" Mixtures," PHYSOR 2004, Chicago, IL (2004).

DiFilippo, F. "Applications of Monte Carlo Methods for the Analysis of the MHTGR Case of the Proteus Benchmark," Oak Ridge National Laboratory, ORNL/TM-12711 (1994).

Donovan, T. "Implementation of Chord Length Sampling for Transport Through a Binary Stochastic Mixture," Proceedings of the American Nuclear Society Topical Meeting in Mathematics and Computation, Gatlinburg, TN, April 2003.

Donovan, T. and Danon, Y., "Application of Monte Carlo Chord-Length Sampling Algorithms to Transport Through a Two-Dimensional Binary Stochastic Mixture," Nuclear Science and Engineering, Vol. 143, 226-239 (2003b).

Donovan, T. Monte Carlo Neutral Particle Transport Through a Binary Stochastic Mixture using Chord Length Sampling, Ph.D. Thesis, Rensselaer Polytechnic Institute (2003).

Doub, W. "Particle Self-Shielding in Plates Loaded with Spherical Poison Particles," Nuclear Science and Engineering, Vol. 10, 299-307 (1961).

Duderstadt, J. and L. Hamilton, Nuclear Reactor Analysis, John Wiley, New York (1976).

Dupree, S. and S. Fraley A Monte Carlo Primer: A Practical Approach to Radiation Transport, Kluwer Academic, New York (2002).

El-Wakil, M. Nuclear Energy Conversion, American Nuclear Society, La Grange Park, IL (1992).

General Atomics, "Development Plan for Advanced High Temperature Coated-Particle Fuels" Report PC-000513, Rev. 0. Accessed 4 Feb. 2008 (2004).

Goeddel, W. "High-Temperature Gas-Cooled Reactor Fuel Materials Development," Materials and Fuels for High-Temperature Nuclear Energy Applications, M.I.T. Press (1964).

Holden, R. Ceramic Fuel Elements, United States Atomic Energy Commission, Gordon and Breach, New York (1966).

Hughes, H. "Uncertainties Beyond Statistics in Monte Carlo Simulations," Radiation Protection Dosimetry, 1-7 (2007).

Hurwitz, H. and P. Zweifel "Self-Shielding of Lumped-Poison Mixtures," Nuclear Science and Engineering, Vol. 1, 438-440 (1956).

Ji, W. *et al*. "Reactor Physics Analysis of the VHTGR Core," Presented at Oak Ridge National Laboratory (2004).

Johnson, J. "Burnup Reactivity and Isotopics of an HTGR Fuel Pebble," Transactions of the American Nuclear Society, Vol. 85, 273-274 (2001).

Kadak, Andrew "A Future for Nuclear Energy—Pebble Bed Reactors" International Journal of Critical Infrastructures, Vol. 1, 330-335 (2005).

Knief, R. Nuclear Engineering: Theory and Technology of Commercial Nuclear Power, 2nd Ed., Taylor and Francis, Washington (1992).

Koberl, O. and R. Seiler, "Detailed Analysis of Pebble-Bed HTR Proteus Experiments with the Monte Carlo Code TRIPOLI4," 2nd International Topical Meeting on High Temperature Reactor Technology, Beijing, China, September 2004.

Kunitomi, Kazuhiko *et al*. "Japan's Future HTR—the GTHTR 300," Nuclear Engineering and Design, Vol. 233, 309-327 (October 2004).

Leppanen, J. "Development of a New Monte Carlo Reactor Physics Code," Helsinki University of Technology, Ph.D. Dissertation (2007).

Lu, S. *et al*. "Combining Stochastic Facies and Fractal Models for Representing Natural Heterogeneity," Hydrogeology Journal, Vol. 10, 475-482 (2002).

MacFarlane, R. "The NJOY Nuclear Data Processing System Version 91," LA-12740-M, Los Alamos National Laboratory (1994).

Malvagi, F. and Byrne, R. "Stochastic Radiative Transfer in a Partially Cloudy Atmosphere," Journal of the Atmospheric Sciences, Vol. 50, 2146-2158 (1992).

Mendius, P. "Criticality Calculations with MCNP: A Primer," Los Alamos National Laboratory, LA-12827 (1994).

Miller, D. "Benchmarks and Models for 1-D Radiation Transport in Stochastic Participating Media," University of California Davis, Ph.D. Dissertation, (2000).

MONK, "Hole Geometry: An Introduction to the Hole Geometry Package as used in MONK and MCBEND," ANSWERS/MONK/REPORT/003 (2003).

Morel, J. and Watson, A. "Homogenized Cross Sections for Statistical Media," Transactions of the American Nuclear Society, Vol. 89, 293-295 (2003).

Murata, I. *et al*. "Continuous Energy Monte Carlo Calculations of Randomly Distributed Spherical Fuels in High-Temperature Gas-Cooled Reactors Based on a Statistical Geometry Model," Nuclear Science and Engineering, Vol 123. 96-109 (1996).

Nickel, H. *et al*. "Long time experience with the development of HTR fuel elements in Germany," Nuclear Engineering and Design**,** Vol. 217, 1-2 141-151 (August 2002).

Olson, G. *et al*. "Chord Length Distributions in Binary Stochastic Media in Two and Three Dimensions," Transactions of the American Nuclear Society, Vol. 89, 307-3090(2003).

Palmer, T. and Q. Wu, "Final Progress Report - Improvements in Neutronics/Thermal Hydraulics Coupling in Two-Phase Flow Systems Using Stochastic-Mixture Transport Methods," DOE NEER Project DE-FC07-00ID13961, April 30, 2003.

Pevey, R. "Techniques for Common PDFs," Course Notes for NE582 Monte Carlo, University of Tennessee Knoxville, http://web.utk.edu/~rpevey/NE582/, accessed January 2008.

Pomraning, G. "Transport in Discrete Stochastic Mixtures," Transport Theory and Statistical Physics, Vol. 27, 405-443 (1998).

Pomraning, G. Linear Kinetic Theory and Particle Transport in Stochastic Mixtures, Vol. 7, World Publishing (1991).

Prinja, A. and E. Fichtl "Atomic Mix Synthetical Acceleration for Transport in Binary Statistical Mixtures," Transactions of the American Nuclear Society, Vol. 92, 734-736 (2005).

Sahni, D "Equivalence of Generic Equation Method and the Phenomenological Model for Linear Transport Problems in a Two-State Random Scattering Medium," J. Mathematical Physics, Vol. 30, Num. 7 (1989).

Samoilov, A.G. *et al*. Dispersion-Fuel Nuclear Reactor Elements, Israel Program for Scientific Translations Ltd., Jerusalem (1968).

Schneider, E. "A Physical and Economic Model of the Nuclear Fuel Cycle," Cornell Unviersity, Ph.D. Dissertation (2002).

Shultis, J. and Faw, R. Radiation Shielding, American Nuclear Society, La Grange Park, IL (2000).

Smith, N. *et al*. "Geometry Modelling and Visualisation for the Monte Carlo Code MCBEND," 8th International Conference on Radiation Shielding, Arlington, TX (1994).

Sutton, T. *et al*. "The Physical Models and Statistical Procedures used in the RACER Monte Carlo Code," KAPL-4840, Knolls Atomic Power Laboratory (1999).

Tang, Chunhe *et al*. "Design and manufacture of the fuel element for the 10 MW high temperature gas-cooled reactor," Nuclear Engineering and Design, Vol. 218, 91-102 (2002)

Torquato, S. Random Heterogeneous Materials: Microstructure and Macroscopic Properties, Springer-Verlag, New York (2002).

Torquato, S., Random Heterogeneous Materials: Microstructure and Macroscopic Properties, Springer-Verlag, New York (2002).

Vanderhaegen, D. Journal of Quantitative Spectroscopy and Radiative Transfer, Vol. 36, 557 (1986).

Widom. S. “Random sequential addition of hard spheres to a volume,” Journal of Chemistry and Physics, Vol. 44, 3888-3894 (1966).

Yamamoto, T. “Extension of cross section homogenization method for particle-dispersed media to layered particles,” Annals of Nuclear Energy, Vol. 33, 804-812 (2006).

Zimmerman, G. and Adams, M. “Algorithms for Monte Carlo Particle Transport in Binary Statistical Media,” Transactions of the American Nuclear Society, Vol. 64, 287-288 (1991).

Vita

Dustin Ray Reinert was born April 10, 1982 in Clifton, Texas to Ray and Susan Reinert. He was raised in Hamilton, Texas and graduated from Hamilton High School in 2000. He enrolled in the University of Texas at Austin and graduated with a Bachelor of Science Degree in Mechanical Engineering with High Honors in 2004. He began graduate study in the Nuclear and Radiation Engineering program in the Department of Mechanical Engineering at the University of Texas at Austin and received a Master of Science Degree in 2006.

Permanent address: 461 W. McCaleb Ln., Hamilton, TX 76531

This dissertation was typed by the author.